INPUT
Publicación práctica para usuarios de
commodore
Febrero 1986 Precio 350 Ptas
Año 1 Número 6

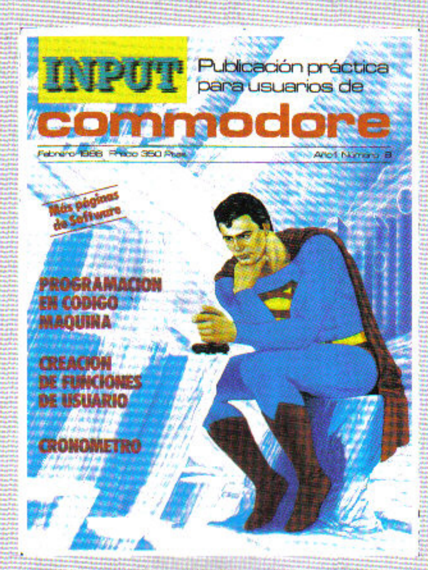
Más páginas de Software

PROGRAMACION EN CODIGO MAQUINA

CREACION DE FUNCIONES DE USUARIO

CRONOMETRO





AÑO 1 NUMERO 6

DIRECTOR:
Alejandro Diges
COORDINADOR EDITORIAL:
Francisco de Molina
DISEÑO GRAFICO:
Tomás López
COLABORADORES:
Luis R. Palencia, Christophe Païs, Francisco Tórtola, Benito Román, Esther de la Cal.

INPUT Commodore es una publicación juvenil de EDICIONES FORUM

GERENTE DIVISION DE REVISTAS:
Angel Sabat

PUBLICIDAD: Grupo Jota
Madrid: c/ Cuenca, 1, 1.º
Teléf. 253 45 01/02
Barcelona: Avda. de Sarriá, 11-13, 1.º
Teléf. 250 23 99

FOTOMECANICA: Ochoa, S. A.

COMPOSICION: EFCA, S. A.

IMPRESION: Edime, S. A.
Depósito legal: M. 27.884-1985

SUSCRIPCIONES: EDISA,
López de Hoyos, 141. 28002 Madrid
Telef. (91) 415 97 12

REDACCION:
Alberto Alcocer, 46, 4.º
28016 Madrid. Teléf. 250 10 00

DISTRIBUIDORA
R.B.A. PROMOTORA DE EDICIONES, S. A.
Travesera de Gracia, 56. Edificio Odiseus.
08006 Barcelona
El precio será el mismo para Canarias que para la Península y en él irá incluida la sobretasa aérea.

Se ha solicitado el control OJD

INPUT commodore

SUMARIO

NUEVOS TEMAS

A pesar de lo sospechado, el mes de febrero parece continuar en la brecha en cuanto a novedades de *software* se refiere. A la redacción continúan llegando nuevos programas de gran calidad.

Los microordenadores están demostrando ser una útil herramienta para los investigadores y curiosos de las nuevas áreas de la informática. Hasta hace poco, temas tales como las aplicaciones robóticas, la síntesis y el reconocimiento de voz, la inteligencia artificial, los sistemas expertos..., etc. aparecían siempre ligados a los grandes ordenadores. Sin embargo, los microordenadores domésticos pueden ayudarnos a comprender los mecanismos de funcionamiento de las aplicaciones que rozan la ciencia ficción. Resulta obvio que la mayor potencia y velocidad de proceso y las ingentes cantidades de memoria central son las principales bazas de los hermanos adultos del micro, pero en los próximos números de INPUT trataremos de acercar esas palabras grandilocuentes a todos vosotros.

A lo largo de la corta existencia de la revista os hemos invitado a participar con vuestras ideas y colaboraciones. Como estímulo complementario venimos ofreciendo tres premios en material informático sorteables entre los participantes del mes. Pues bien, en la próxima edición publicaremos el nombre de los tres ganadores de los tres primeros meses. Así que estar atentos.

Una aclaración, hemos recibido alguna que otra carta de lectores que nos «han pillado» en *off-side* con las páginas centrales de la revista. En realidad la cuestión es otra. Las ocho páginas del cuadernillo central forman un coleccionable, que pretende enseñar el lenguaje BASIC y, en un próximo futuro, código máquina a través de la programación de juegos, ganando en complejidad a medida que avanzamos. El fin último es que dichas páginas sean extraídas y encuadernadas con las tapas que hemos preparado. Por ello es lógico que la numeración de las páginas dé un salto al llegar a este punto e, incluso, que el texto de un capítulo necesite más espacio del disponible, continuando en el siguiente número.

Una recomendación: Seguir enviando vuestras colaboraciones inéditas.

LOS MEJORES DE INPUT

Hemos pensado que es interesante disponer de un ***ranking*** que ponga en claro, mes a mes, cuáles son los programas preferidos de nuestros lectores. Para ello, es obligado preguntaros directamente y tener así el mejor termómetro para conocer vuestras preferencias. Podéis votar por cualquier programa aunque no haya sido comentado todavía en **INPUT**.

El resultado de las votaciones será publicado en cada número de **INPUT**.

Entre los votantes sortearemos 10 cintas de los títulos que pidáis en vuestros cupones.

Nota: No es preciso que cortéis la revista, una copia hecha a máquina o una simple fotocopia sirven.

Enviad vuestros votos a: **LOS MEJORES DE INPUT** Alberto Alcocer, 46 - 4.º B. 28016 Madrid

ELIGE TUS PROGRAMAS

Primer título elegido ____________________ **Segundo título elegido** ____________________

Tercer título elegido ____________________ **Programa que te gustaría conseguir** ____________________

Qué ordenador tienes ____________________ **Nombre** ____________________

1.er Apellido ____________________ **2.º Apellido** ____________________

Fecha de nacimiento ____________________ **Teléfono** ____________________

Dirección ____________________ **Localidad** ____________________

Provincia ____________________

INPUT COMMODORE N.º 6

EL EVOLUCIONADO SOFTWARE AMERICANO

Con pantallas, instrucciones y presentación en:

CASTELLANO

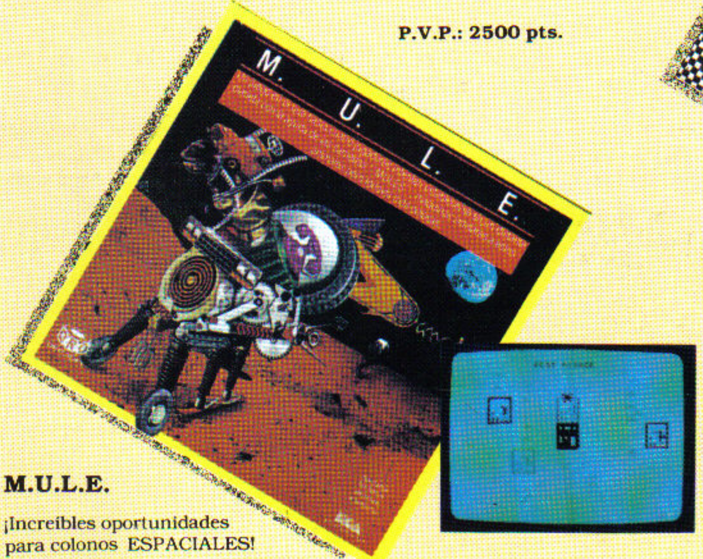


REINO DE LO IMPOSIBLE

¡DIOS MIO! ¡SE ACERCAN!

Ellos son miles y vosotros sólo dos. ZOMBIES, ARAÑAS, SERPIENTES y toda clase de pesadillas os impiden escapar del Reino de lo Imposible. ¡COMO ALCANZAR LA LLAVE! Un increíble juego de colaboración entre dos jugadores luchando por su vida en un extraño laberinto terrorífico. ¿Conseguirán alcanzar la puerta de la realidad mientras el sólo contacto de los ZOMBIES les arrebata la vida poco a poco?

P.V.P.: 2500 pts.



M.U.L.E.

¡Increíbles oportunidades para colonos ESPACIALES!

GRANDES FACILIDADES de Explotación y Desarrollo Industrial en la superficie indómita de un Planeta paradisiaco.
¡Subastas de terrenos!
¡Abundante producción de SMYTHORE y KRISTITA!
¡Pocas probabilidades de sufrir el ataque de los PIRATAS Espaciales!
¡Una oportunidad de URANIO para su familia!
Un juego de economía planetaria en un futuro distante.

HASTA CUATRO JUGADORES. **P.V.P.: 2500 pts.**

RACING DESTRUCTION SET

El mejor juego de carreras de coches jamás inventado

Igual que en los juegos de coches eléctricos, diseña con tu ordenador TU PISTA IDONEA.

Elije tu coche.

Selecciona entre varios tipos de tramos y ensámblalos.

Elije el terreno o las mezclas de terreno... tierra, asfalto, hielo.

Peralta las secciones que desees.

Y ¡COMPITE! en una carrera tête a tête con tu enemigo del volante.

DOS JUGADORES con DOS PANTALLAS SUBJETIVAS

Saltos mortales y... juego avanzado de combate en la autopista.

Si no puedes adelantarle, al menos podrás ARRANCARLE LAS CUATRO RUEDAS AL CHASIS.

P.V.P.: 2500 pts.

Continuamos desarrollando nuestro Catálogo.



DRO SOFT

Editado por DRO SOFT. Fundadores, 3 - 28028 MADRID
Tlfs.: 255 45 00/09



ELECTRONIC ARTS



NUEVOS PRECIOS

Microelectronica y Control, firma representante de Commodore ha recomendado un reajuste de precios de venta a sus distribuidores.
De acuerdo con ello, el C-64 tendrá un precio de 44.500 pts, el C-128 costará 71.300 pts, el Personal Computer 353.000 pts y la unidad de *diskettes* 1541 pasa a 41.900 pts.
La firma justifica la nueva tarifa en función de la desaparición del impuesto arancelario de 15.000 pts y la aplicación del I.V.A.



MAS MEMORIA PARA LOS MICROS

El Comdex es la mayor feria de informática y se celebra anualmente en Las Vegas (EE.UU.). Allí dos firmas japonesas, Hitachi y Toshiba, acaban de presentar sus modelos de almacenamiento masivo en disco (Compact Disk) capaces de guardar hasta 3.600 Megabytes una y 2.600 Megabytes la otra. Sorprendente!!!.
En su momento, cualquier microordenador podrá disponer de la información depositada en uno de esos discos ópticos de 12 pulgadas de diámetro. La lectura se efectúa por medio de un diminuto sistema láser.
La principal desventaja de estos sistemas es que actúan como ROM, es decir únicamente puede leerse la información depositada a *priori*.
Varias firmas japonesas presentaron otros sistemas de disco compato que pueden grabarse una sola vez, siendo una especie de PROM (ROM programable).



DE NUEVO EL 128

El C-128 se esta ofreciendo en Gran Bretana al precio de 450 Libras incluyéndose en el precio la unidad de disco.
Esta nueva unidad, según el fabricante, es capaz de leer *diskettes* con programas desarrollados bajo el sistema operativo CP/M en los formatos empleados por IBM, Kaypro y Osborne, pudiendo aprovechar el *software* escrito en su día para estos ordenadores.



OTRA VEZ AL CINE

Domark, firma de *software* que adquiriese popularidad en su día por el programa basado en la película "Panorama para matar", ha lanzado un nuevo título con tema cinematográfico: "Viernes 13".
El juego se centra en proteger a los quinceañeros de las garras del malvado Jason, que con hacha y sierra mecánica los atacará sin piedad.
Desconoces dónde se encuentra el villano de la máscara de *hockey*, pero cuentas con diez armas y cinco niveles para cazarle. Es un juego de gran suspense.
La cinta viene acompañada por dos cápsulas de sangre artificial y existirá la posibilidad de ganar un monitor con sólo reconocer a qué corresponden los sonidos grabados en la cinta.
El sonido incluido en el programa contribuye a darle mayor dramatismo.
La casa Serma lo importará en nuestro país.



INCREIBLE AMIGA

Recientemente se hizo una presentación en petit comite de un simulador de vuelo para el potente microordenador Amiga, en el que se incluían elementos característicos de los juegos de marcianitos, llamado Radar Raiders.
Un programa tan alucinante por sus gráficos y sonidos que nadie podía creer que el ejecutante disponía de tales capacidades.



DIGITALIZADOR PARA EL COMMODORE PC

Ya es posible digitalizar imágenes de video con el nuevo sistema de Print Technik conectado al Personal Computer de Commodore.
Se trata de un dispositivo alojado en el interior de una pequeña caja de plástico, por la que asoma un cable que termina en un conector que se enchufa en el *port* destinado a la impresora paralelo.
La resolución es de 256 por 256 *pixels*, proporcionando hasta 16 colores.
La fuente de señal puede ser cualquier dispositivo capaz de generar una señal de video: una camara de tv, un video, etc..
Las imágenes digitalizadas pueden ser almacenadas en *diskette* o incluso impresas en papel.
Es preciso que el ordenador disponga de la tarjeta para gráficos en color.

CONSTRUYE TU CIRCUITO IDEAL

EN CASTELLANO

Igual que en los juegos de coches eléctricos, diseña con tu ordenador TU PISTA IDONEA.
Elije tu coche.
Selecciona entre varios tipos de tramos y ensámblalos.
Elije el terreno o las mezclas de terreno... tierra, asfalto, hielo.
Peralta las secciones que desees.
Y ¡COMPITE! en una carrera tête a tête con tu enemigo del volante.

Racing DESTRUCTION Set

DOS JUGADORES con DOS PANTALLAS SUBJETIVAS

Saltos mortales y...
juego avanzado de combate en la autopista.
Si no puedes adelantarle, al menos podrás ARRANCARLE LAS CUATRO RUEDAS AL CHASIS.
Esto es:
RACING DESTRUCTION SET

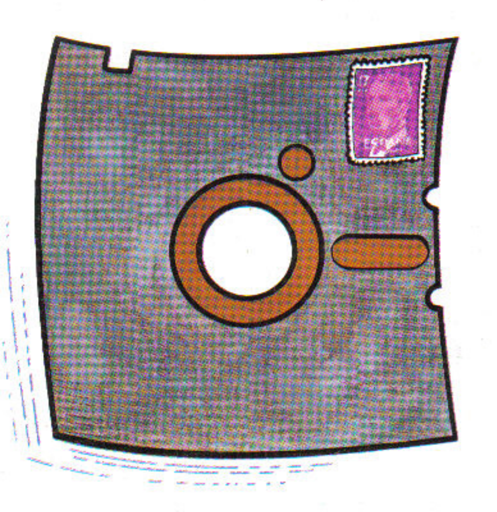
P.V.P.: 2500 pts.

La empresa de Software que en menos de dos años se ha convertido en líder absoluto del mercado norteamericano por la alta creatividad y sofisticación de sus programas producidos y presentados ahora en España por D.R.O. SOFT, completamente traducidos al Castellano para Commodore, Spectrum y Amstrad.





Editado por DRO SOFT. Fundadores, 3 - 28028 Madrid.
Telfs: 255 45 00 / 09



EL BUZON DE INPUT

Respuesta a Luis Díez de Tejada. Madrid:

No entra en la línea editorial de esta publicación polemizar en torno a los múltiples problemas que pueden surgir con los servicios técnico oficiales, muchos de los cuales conocemos por otras cartas que con mayor o menor asiduidad llegan a esta redacción. Esto no se debe a que nos ate tipo alguno de compromiso con el fabricante o sus representantes, sino todo lo contrario; no estamos en situación de poder remediar nada. A ese nivel sólo somos un usuario más, que se diferencia de vosotros por el hecho de hacer revistas.

No se puede negar que tu dossier viene totalmente documentado. Sin embargo, en la fotocopia del albarán aparece un concepto que no deja de producirnos, cuando menos, hilaridad. Nos referimos al «Ajuste frecuencia cristal». Pues bien, un ***Commodore 64*** *incorpora un cristal de cuarzo cuya frecuencia de resonancia es de 17,734472 MHz en el caso de trabajar en norma PAL de televisión. Su función es crear una señal de «reloj» altamente estable frente a cambios de tensión y/o temperatura entre otras cosas. Esto es de importancia vital, pues a partir de él se producen las señales de sincronismo y generación de la imagen que veremos en la pantalla. También actúa como reloj del microprocesador. Este cristal puede observarse en el poster que publicamos con el primer número de la revista, es la cajita metálica que aparece tumbada encima del chip marcado con un 6. En esta foto se han sustituido una serie de chips por otro fabricado por* ***MOS Tecnologies,*** *firma dependiente de* ***Commodore,*** *pero en el esquema original de 1982 el cristal oscilaba con la ayuda del circuito integrado 74LS629N, que por ser un oscilador controlable por tensión (VCO) permite forzar pequeñas variaciones en la frecuencia propia del cristal con sólo girar un potenciómetro ajustable de 2 Kbytes con un simple destornillador. Esta operación podría definirse como «ajuste de la frecuencia del oscilador patrón» o similar. Una operación que no habría de llevar más de cinco minutos con la ayuda de un modesto frecuencímetro de alta frecuencia (valga la redundancia).*

Para que comprendas nuestra perplejidad, te explicamos qué es un cristal de cuarzo de este tipo. Partiendo de un determinado cuarzo natural, se obtiene una lámina siguiendo una orientación cristalográfica adecuada, que permitirá que el cristal oscile como respuesta una señal eléctrica excitadora aplicada a sus caras. La frecuencia marcada por el cristal y es función de varios parámetros, entre los cuales sus dimensiones fundamentalmente el espesor tienen importancia primordial. Se dispone la oblea de cuarzo en un circuito oscilante y se observa su frecuencia de resonancia, ajustándose a la frecuencia deseada puliendo mecánicamente sus caras para disminuir el grosor. En la actualidad el proceso puede realizarse con ayuda de la química. Posteriormente se metalizan ambas caras por evaporación de un metal conductor, convirtiéndose en una especie de condensador, en el que el dieléctrico es el propio cuarzo. Se añaden los terminales por soldadura y se encapsula y sella todo el conjunto, para aislarlo de los golpes y los cambios de temperatura. Ajustar literalmente la frecuencia del cristal requiere desencapsular el cristal, pulirlo, metalizarlo..., demasiado caro y complejo, es un componente no muy cro y si estuviera dañado sería preferible cambiarlo por otro.

En fin, estamos de acuerdo en que frente a términos tan modestos como fusible o conector, el aludido ajuste suena muy rimbombante.



Os escribo para preguntaros dónde puedo comprar u obtener información sobre el reloj Seiko RC-100, **publicado en el número 1 de la revista y, si es posible, decid también las ventajas que puede suponer la utilización de la unidad** 1571 **con respecto a la 1541 en un** C-64 **si es que la tiene.**

Miguel Angel Tartaj. Zaragoza.

El interesante reloj terminal ***RC-100*** *de* ***Seiko*** *apareció en la sección de Actualidad de la revista por haber sido presentado en el* ***Commodore Show*** *que se celebró poco tiempo antes en Londres. En Inglaterra ya está comercialmente disponible, pero no tenemos constancia de que lo esté en nuestro país. Esa decisión compete al importador, que marca sus estrategias de producto en función de criterios propios. Es por ello que desconocemos en qué momento pueda comprarse aquí y si lo encontrarás en relojerías o en tiendas de microinformática. De cualquier modo, quienes podáis estar interesados en él no dudéis en consultárselo al representante de la marca, seguramente anticipará su comercialización cuando comience a tener muchas llamadas de usuarios interesados.*

La unidad de diskettes 1571 fue concebida como compañero para el anunciado ***Commodore 128*** *y se pusieron muchas espectativas en su llegada. Su principal ventaja se centra en la mayor velocidad de transferencia de datos, disminuyéndose de esta forma el tiempo necesario para guardar o almacenar programas. Sin embargo, por los comentarios que nos llegan de* →

EL BUZON DE INPUT

países europeos en los que ***Commodore*** *tiene una importante presencia, la unidad no ha salido de las cadenas de montaje todo lo madura que sería de suponer.*



Tengo un Commodore 16, **y cuando quiero hacer un programa tengo que pedir a un amigo qué significa cada POKE...**

Javier Fernández. Alicante.

Me gustaría que publiquéis, si no es mucha molestia, una pequeña tabla con las direcciones de memoria utilizadas con las instrucciones ***POKE y a la vez aparecieran*** **las equivalencias en un** Commodore 16. **Me refiero a la memoria de pantalla, a la de cambio de color de fondo, etc.**

Luis Angel Carlar.
Castejón. Navarra.

Hemos recibido unas cuantas cartas vuestras, usuarios del ***Commodore 16****. No creáis que os tenemos en un olvido premeditado, es más, estamos haciendo esfuerzos para localizar autores que puedan preparar artículos y adaptaciones para él. Sin embargo, no son fáciles de encontrar, y lejos de intentar desanimaros, se podría afirmar que el* ***Commodore 16****, al igual que el* ***PLUS 4*** *(en España sólo se vendió a través de canales de importación paralelos) son dos de los hijos huérfanos de* ***Commodore****. A pesar de todo dispone de una envidiable versión del BASIC, pero el fabricante siempre ha venido huyendo de la compatibilidad entre sus máquinas, periféricos.*

Concretamente, en nuestro país se vendieron unos pocos miles de ***C-16*** *(otros se sortearon con productos de pastelería y afines), pero no hemos observado esfuerzo alguno para su promoción a nivel usuario. Es de suponer que cuando uno compra un ordenador, un automóvil, un electrodoméstico, etc., el fabricante y sus distribuidores adquieren el compromiso, al menos moral, de proporcionar todo tipo de soporte postventa a sus usuarios. En el caso de los ordenadores, deberían animar a los fabricantes de software, accesorios y periféricos a lanzar una gama que permita aprovechar el ordenador al máximo. Nosotros mismos tenemos un* ***C-16*** *en la redacción y su correspondiente manual, pero ni una línea más de información relativa al mismo. Por vuestro lado, lo más aconsejable sería que os pusierais en contacto con el distribuidor nacional, para que dé respuesta a vuestras inquietudes. Nosotros por nuestro laso continuaremos buscando soluciones, par ayudaros todo lo que esté en nuestras manos. No lo dudéis, hacer llegar vuestras quejas.*



---

países europeos en los que ***Commodore*** *tiene una importante presencia, la unidad no ha salido de las cadenas de montaje todo lo madura que sería de suponer.*



Tengo un Commodore 16, **y cuando quiero hacer un programa tengo que pedir a un amigo qué significa cada POKE...**

Javier Fernández. Alicante.

Me gustaría que publiquéis, si no es mucha molestia, una pequeña tabla con las direcciones de memoria utilizadas con las instrucciones *POKE y a la vez aparecieran* las equivalencias en un Commodore 16. **Me refiero a la memoria de pantalla, a la de cambio de color de fondo, etc.**

Luis Angel Carlar.
Castejón. Navarra.

Hemos recibido unas cuantas cartas vuestras, usuarios del ***Commodore 16****. No creáis que os tenemos en un olvido premeditado, es más, estamos haciendo esfuerzos para localizar autores que puedan preparar artículos y adaptaciones para él. Sin embargo, no son fáciles de encontrar, y lejos de intentar desanimaros, se podría afirmar que el* ***Commodore 16****, al igual que el* ***PLUS 4*** *(en España sólo se vendió a través de canales de importación paralelos) son dos de los hijos huérfanos de* ***Commodore****. A pesar de todo dispone de una envidiable versión del BASIC, pero el fabricante siempre ha venido huyendo de la compatibilidad entre sus máquinas, periféricos.*

Concretamente, en nuestro país se vendieron unos pocos miles de ***C-16*** *(otros se sortearon con productos de pastelería y afines), pero no hemos observado esfuerzo alguno para su promoción a nivel usuario. Es de suponer que cuando uno compra un ordenador, un automóvil, un electrodoméstico, etc., el fabricante y sus distribuidores adquieren el compromiso, al menos moral, de proporcionar todo tipo de soporte postventa a sus usuarios. En el caso de los ordenadores, deberían animar a los fabricantes de software, accesorios y periféricos a lanzar una gama que permita aprovechar el ordenador al máximo. Nosotros mismos tenemos un* ***C-16*** *en la redacción y su correspondiente manual, pero ni una línea más de información relativa al mismo. Por vuestro lado, lo más aconsejable sería que os pusierais en contacto con el distribuidor nacional, para que dé respuesta a vuestras inquietudes. Nosotros por nuestro laso continuaremos buscando soluciones, par ayudaros todo lo que esté en nuestras manos. No lo dudéis, hacer llegar vuestras quejas.*



EXIGELE A TU BASIC

- HAZ QUE TUS PROGRAMAS EN BASIC CORRAN MAS
- TIEMPO DE EJECUCION DE LAS SENTENCIAS
- ATENCION A LA ESTRUCTURA

Ponte el casco, la visera y las ropas de viaje. Limpia tus bujías y asegúrate de que tu máquina corre con suavidad. Está a punto de bajarse la bandera y ya es tiempo de acelerar tu BASIC.

Al principio, todos los ordenadores parecían rápidos como un rayo, realizando «en un abrir y cerrar de ojos» tareas que a cualquier persona le requerirían mucho más tiempo. Pero prueba a escribir un juego de acción, o hacer que la máquina ejecute una larga serie de cálculos, o un trabajo complejo de clasificación, con el BASIC y te encontrarás con una situación muy distinta. Verás que realmente la máquina tarda en terminar la tarea que se le ha asignado. Puede que muy pronto empieces a quejarte de lo lenta que es.

Los programas que se ejecutan con más rapidez son los que están escritos en código máquina o lenguaje ensamblador, si bien la mayor parte de la gente encuentra más fácil programar en BASIC. Desgraciadamente, un programa en BASIC nunca puede esperar ni siquiera acercarse a la velocidad de un programa escrito en código máquina, debido a que el ordenador tiene que gastar tiempo traduciendo las instrucciones del BASIC al código máquina. Para ello lleva incorporado un programa especial que se ocupa de este trabajo: el intérprete.

Si tú no quieres escribir programas en código máquina, pero sí apurar al máximo la velocidad de tu máquina, puedes ensayar varios trucos. En primer lugar, procura estructurar adecuadamente tus programas. En segundo lugar, tienes que intentar conseguir que cada línea individual de programa opere a la velocidad óptima. Elige



pues los componentes del BASIC que el intérprete traduce con más rapidez.

Cada máquina tiene sus propias peculiaridades, y en cierta medida, cada programa tiene sus propias exigencias. En consecuencia no existen reglas generales e infalibles para conseguir el programa perfecto. Sólo se pueden dar ciertas indicaciones, y a ti te corresponde escoger los trucos a incorporar a tus programas, ya que el uso de los mismos puede requerir sacrificar otras cosas.

MEDIDA DE TIEMPOS EN PROGRAMAS DE BASIC

Todas las máquinas llevan incorporado un reloj que puede utilizarse para comparar la rapidez de unos programas respecto a otro. Teclea la rutina siguiente, especialmente diseñada para tu máquina, para que puedas ir viendo por tí mismo en los ejemplos posteriores de este artículo las diferencias entre formas alternativas de programar en BASIC:

Teclea para Commodore 64

```
100 T=TI
110 FOR I=1 TO 100:GOSUB 200
    :NEXT I
120 T=TI-T
130 PRINT"[SHIFT+CLR/HOME]
    TIEMPO TOMADO=";(T-21)/6
    ;" MILISEGS."
140 END
200 REM
500 RETURN
```

Teclea para Vic-20

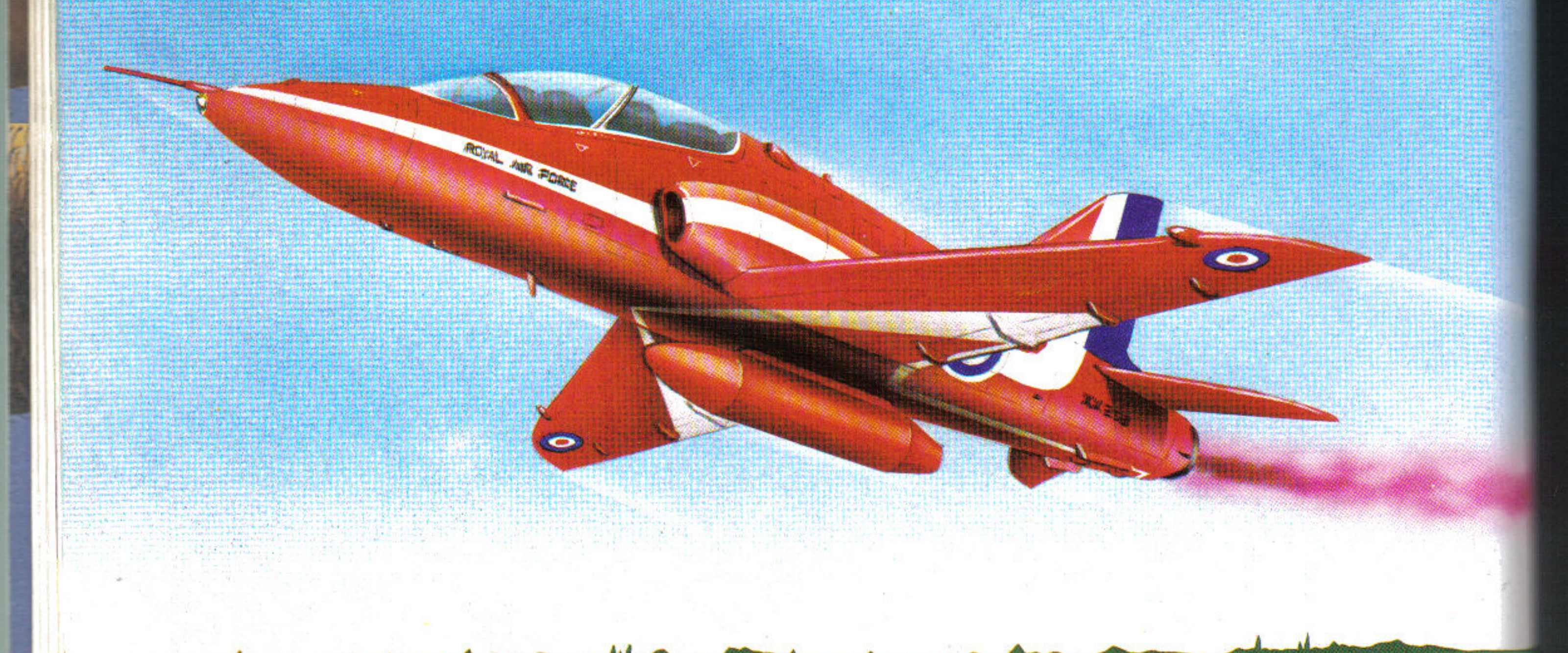
```
100 T=TI
110 FOR I=1 TO 100:GOSUB 200
    :NEXT I
120 T=TI-T
130 PRINT"[SHIFT+CLR/HOME]
    TIEMPO TOMADO=";(T-18)/6
    ;" MILISEGS."
140 END
200 REM
500 RETURN
```

Observa la sentencia REM en la línea 200, que será utilizada posteriormente para encajar con la parte que se quiera probar. La sentencia REM de la línea 100 forma parte de la corrección de cronometraje y no debe ser omitida.

ESTRUCTURA

Aunque ya hemos hablado bastante en INPUT sobre los programas estructurados, merece la pena insistir en aquellos factores que más influyen en la velocidad de ejecución.

Todas las subrutinas que se utilicen con frecuencia, tienes que situarlas cerca de principio del programa. La razón de esto es que el intérprete empieza a buscar el número de la línea que aparece junto al GOSUB desde el principio. Obviamente, si el número de línea es bajo, será encontrada con



más rapidez que si es alto. El ahorro de unos milisegundos aquí y allá puede convertirse enseguida en una contribución significativa en cuanto a velocidad.

Los programas mal pensados que utilizan un laberinto indiscriminado de GOTOs —a los que a veces se llama programas de *spaghetti*— no sólo constituyen una seria amenaza para los que intenten leer el programa, sino que pueden ser un gran obstaculo para la ejecucción rápida. Asi pues, un programa bien planeado es probable que sea más rápido de ejecución que uno que has ido modificando una y otra vez sin un plan previo.

MEMORIA Y VELOCIDAD

En general, los programas cortos es muy probable que se ejecuten también más rápidamente. Sin embargo, las tres condiciones principales que debe cumplir un programa —velocidad, claridad y economía de memoria— suelen estar normalmente en conflicto. Por ejemplo, las líneas de programa con muchas sentencias, sirven para ahorrar memoria y aumentar la velocidad, pero pueden hacer que el listado sea más difícil de seguir y depurar.

Un programa diseñado para tener máxima velocidad de ejecución, es posible que aumente mucho en longitud y en uso de memoria. Recíprocamente, algunos programas que son muy efectivos para ahorrar memoria, lo consiguen a expensas de un aumento de velocidad.

Se puede ahorrar mucha memoria utilizando subrutinas, pero las llamadas a subrutinas consumen tiempo y por lo tanto son indeseables si lo que quieres es una ejecución muy rápida. Como ya has visto, si optas por las subrutinas, tienes que ser muy cuidadoso con la estructura del programa para poder recuperar algo de la velocidad perdida.

Análogamente, LET A = VAL «100» está muy bien para ahorrar memoria, pero resulta desastrosa cuando se compara con LET A = 100, que consume más memoria, pero está más indicada para aumentar la velocidad.

VARIABLES

Si quieres acelerar tus programas, te resultará muy útil el conocimiento de cómo se almacenan las variables en la memoria. El área de variables se limpia con RUN o CLEAR, y las variables se crean a medida que van apareciendo. Generalmente las nuevas variables se añaden extendiendo hacia arriba el área de variables.

Consideremos una situación en la que se crea una variable que es una cadena de caracteres y a continuación una matriz númerica. Después puedes añadir algo a la cadena de caracteres. Si la cadena ya había sido creada antes en el programa, todas las variables creadas con posterioridad tendrán que desplazarse hacia arriba en la memoria, para acomodarse a las condiciones de la cadena.

Echa un vistazo a este ejemplo:

```
100 LET T$=""
110 DIM A(1000)
120 LET T$=T$+"LO QUE SEA"
```

En casi todas las versiones de BASIC se puede ahorrar bastante tiempo permutando las dos primeras líneas de programa, es decir, dimensionando la matriz antes de definir la cadena de caracteres. Tal como está el programa, puede que haya que mover 5000 bytes cada vez que se suma «LO QUE SEA» a T$.

Hay una regla general, por la que si usas variables en vez de números, ahorrarás gran cantidad de tiempo. Por





P y R

¿Por qué estas reglas de velocidad se aplican solamente a algunas máquinas?

No hay normas rígidas sobre cómo escribir los programas en BASIC para arrancarle el esfuerzo supremo de velocidad, como tampoco las hay sobre la forma de escribir intérpretes; después de todo son programas de máquinas. Pueden contener compromisos parecidos a los de los programas que tú puedas escribir.

La velocidad de un intérprete dependerá de la forma en que esté escrito y de las prestaciones que el fabricante quiera incorporar en su BASIC.

P y R

El BASIC me sigue pareciendo demasiado lento, pero creo que programar en código máquina es muy tedioso. ¿Qué otra cosa puedo hacer?

Hay una especie de término medio, constituido por los lenguajes compilados en vez de interpretados. Lo que más tiempo requiere es ir interpretando el BASIC mientras el programa está corriendo.

El programa compilado utiliza en cambio otro tipo de programa —parecido a los familiares ensambladores de la programación en código máquina— que convierte el programa escrito en un lenguaje de alto nivel en un programa en código máquina, antes de empezar la ejecución. Esto significa que después de la compilación, se ejecuta un programa en código máquina en vez de un programa de alto nivel.

ejemplo, en el **Commodore 64** cada variable ahorra aproximadamente de cinco a diez milisegundos. Puede que esto no te parezca gran cosa, pero si hay muchos bucles en los que constantemente se manejan números, el tiempo total ahorrado podría ser considerable.

FUNCIONES MATEMATICAS

Utiliza las rutinas de medida de tiempos para comparar estas dos maneras de calcular la misma cosa:

```
200 LET C=4*4*4*4
```

o bien

```
200 LET C=4^4
```

A lo mejor te creías que las funciones que la máquina lleva incorporadas resultan de ejecución más rápida. He aquí otro ejemplo del conflicto entre economía y velocidad de ejecución.

Se ha sugerido a veces líneas como la siguiente:

```
210 IF X>Y THEN Y=Y+1
220 IF X<Y THEN Y=Y-1
```

pueden ganar mucho en velocidad sustituyéndolos por:

```
210 Y=Y+(X>Y)-(X<Y)
```

Prueba en ambos casos con diversos valores de X e Y.

MULTIPLICACION Y DIVISION

Las expresiones C = D*0.5 y C = D/2 realizan exactamente el mismo cálculo, pero te encontrarás que la multiplicación es algo más rápida.

Prueba las siguientes sugerencias para ver cuál es más rápida y anota los resultados por si te hacen falta en el futuro:

Teclea para C-64 y Vic-20

```
200 LET C=10+10
200 LET C=D+D(donde D=10)
```



```
200 LET C=10*10
200 LET C=10/10
200 LET C=10+[PI]
200 LET C=SIN (10)
200 LET C=COS (10)
200 LET C=TAN (10)
200 LET C=VAL ("10")
200 LET C=10
200 LET C=D(donde D=10)
200 PRINT "TEST"
200 PRINT A$(donde A$="TEST")
200 PRINT 10+1000+500+5.5
200 PRINT D+E+F+G(donde D=10
     ,E=1000, etc.)
```



CLASIFICACION Y BUSQUEDA

En un próximo artículo de INPUT nos ocuparemos en profundidad del tema de la clasificación. La ordenación de **Shell-Metzner**, que veremos en dicho artículo, sería la más adecuada para utilizar en cualquier aplicación que implicara la ordenación de, por ejemplo, más de cien elementos. Tiene la particularidad de que cuanto más datos tienes que manejar, más rápida es clasificando. Es un método mucho más rápido que el conocido método de la burbuja, aunque es también un ejemplo de sacrificio de memoria en aras de una mayor velocidad.

Muchas veces se considera el problema de la búsqueda junto con el de la ordenación. En este caso, en vez de poner en orden una colección de datos, lo que se pretende es recuperar un elemento en particular (o un conjunto de elementos relacionados) con la mayor rapidez posible. Suponte que tienes en tu ordenador una lista de números de teléfono, y que necesitas el número de Alberto Gómez e Hijos, Papeles Pintados. Naturalmente lo que quieres es que la máquina tarde menos en encontrar el número que si tienes que buscarlo en la guía de teléfonos.

La búsqueda en serie, hace lo mismo que una persona que recorre una lista en un papel, nombre a nombre. El listado es el siguiente:

Teclea para C-64 y Vic-20

```
10 REM**BUSQUEDA EN SERIE
20 DIM B$(10)
```

REGLAS PARA AUMENTAR LA VELOCIDAD DEL BASIC

- Empieza las matrices a partir del subíndice 0 y no a partir del 1.
- En los bucles FOR ... NEXT, no pongas variables después del NEXT.
- Define las matrices al principio del programa.
- Siempre que sea posible, utiliza variables enteras en vez de con valores decimales.
- Utiliza variables en vez de números.
- Utiliza nombres cortos para las variables. Siempre que sea posible, una sola letra (también **Vic 20**).
- Coloca todas las subrutinas de uso frecuente al principio del programa (también **Vic 20**).
- Utiliza bucles FOR ... NEXT con preferencia a bucles del tipo
100 LET X = X+1: IF X < 20 THEN GO TO 100
- Usa números pequeños para las líneas. Empieza a programar a partir de la línea 10 y no a partir de la 1000 (también **Vic 20**).
- Emplea las cortas rutinas en código máquina a las que se llama desde el BASIC para el *scrolling*, etc. Conviértete en un ávido coleccionista de las rutinas que se publican en código máquina (también **Vic 20**).
- Planifica tu programa sobre el papel antes de empezar a teclear. De esta forma evitarás la pérdida de tiempo que suponen los excesivos saltos arriba y abajo, que son síntomas de un programa mal planeado (también **Vic 20**).
- Recuerda que las subrutinas ahorran memoria pero gastan tiempo (también **Vic 20**).
- Elimina todos los denominadores comunes superfluos. Sustituye
10 LET X = Y/100 + Z/100 por
10 LET X = (Y + Z)/100 (también **Vic 20**).
- Evita la sentencia GO TO siempre que sea posible (también **Vic 20**).
- Siempre que puedas, escribe varias sentencias en una línea (también **Vic 20**).
- Elimina del programa todos los espacios innecesarios, las líneas en blanco y las sentencias REM (también **Vic 20**).
- En las sentencia IF, coloca primero las condiciones con más probabilidades de ser falsas (también **Vic 20**).
- Reutiliza los nombres de variables y las variables de bucle, en vez definir otros nuevos.

```
30 PRINT"[SHIFT+CLR/HOME]
   [3*CRSR abajo]":FOR I=1
   TO 10:READ B$(I):PRINT I,
   B$(I):NEXT I
40 INPUT "[CLR/HOME]ESCRIBE
   LA PALABRA[CRSR izq.]";A$
   :PRINT "[CLR/HOME][3*CRSR
   abajo]"
50 FOR X=1 TO 10
60 IF B$(X)=A$ THEN PRINT
   SPC(4);"[CTRL+9]HALLADA":
   GOTO 75
70 PRINT:NEXT X
75 PRINT "[CLR/HOME]"
80 DATA "ANTILOPE","BURRO",
   "CABRA","DELFIN",
   "ELEFANTE"
85 DATA "GORRION","IGUANA",
   "KOALA","LEON","PERRO"
```

Aunque la búsqueda serial es una rutina muy conocida, no es muy rápida. La búsqueda binaria o dicotómica no es mucho más difícil de programar, pero es de ejecución mucho más rápida, si bien la diferencia se advierte mejor cuando las listas son largas.

La mayor velocidad de la búsqueda binaria se debe a que, a diferencia de la búsqueda en serie, no tiene que examinar todos los elementos de la lista. Previamente los datos tienen que estar ordenados por orden alfabético o numérico, y el ordenador mira primero al elemento que hay en el centro de la lista. Desde este punto se mueve hacia arriba o hacia abajo, reduciendo a la mitad cada vez la lista restante, ya que va comparando cada elemento que se encuentra con el que va buscando. Inicialmente no busca un coincidencia perfecta, como en la búsqueda serie, sino que simplemente observa si la primera letra es más alta o más baja que la del elemento deseado.

Teclea para C-64 y Vic-20

```
10 PRINT"[SHIFT+CLR/HOME]"
20 T%=10:B%=1
30 DIM N$(10)
40 FOR C=1 TO 10
50 READ N$(C)
60 NEXT C
70 INPUT"ANIMAL BUSCADO "
   ;A$
80 TI$="000000"
90 IF TI<50 THEN 90
95 PRINT"[SHIFT+CLR/HOME]"
100 IF N$(T%)=A$ THEN PRINT
    N$(T%),T%: GOTO 200
110 IF N$(B%)=A$ THEN PRINT
    N$(B%),B%: GOTO 200
120 P%=(T%+B%)/2
130 IF N$(P%)=A$ THEN PRINT
    N$(P%),P%: GOTO 200
140 IF N$(P%)>A$ THEN T%=P%
150 IF N$(P%)<A$ THEN B%=P%
160 IF T%-B%=1 THEN PRINT
    "NO HALLADO": GOTO 200
170 GOTO 100
200 GET Y$:IF Y$="" THEN 200
210 RUN
580 DATA "ANTILOPE","BURRO",
    "CABRA","DELFIN",
    "ELEFANTE"
585 DATA "GORRION","IGUANA",
    "KOALA","LEON","PERRO"
```

Para acelerar los programas de BASIC no hay una respuesta única. Los mejores resultados se obtienen con una atención cuidadosa a muchos detalles aparentemente insignificantes y con mucha experimentación. Como cada programa es un caso especial, nunca puede haber reglas infalibles sobre la forma en que hay que escribir programas. Siempre hay cosas nuevas por descubrir y en parte a ello se debe la fascinación de la programación. Es muy probable que los mayores aumentos de la velocidad de ejecución de tus programas se deban a tus propios descubrimientos.

Ahora exclusiva para España **Bristish Soft.**
Instrucciones en castellano. Importados de Inglaterra. Todos los precios incluido IVA. Tiendas y distribuidores. Tel. (965) 26 35 93. Pedidos contra-reembolso. Tel. (965) 26 35 93. Disponible directamente por British Soft o en los mejores establecimientos de software.
British Soft. Rocafel, 19. Albufereta *(Alicante)*

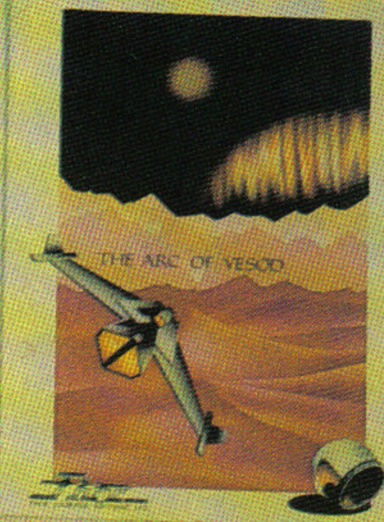
Distribuidor en Madrid:
Computique
Embajadores, 90
28012 Madrid
Tel. (91) 227 09 80



Icucs: de THOR Computer Software. Nuevo. El juego más reciente de Odin. Terminado de fabricar en enero. Para Spectrum 48, 128 K. Commodore 64 ó 128 K. P.V.P. **2.300**



Robin The Wood: de Odin Computer Graphics. Crash Mash nuevo para Amstrad. ¿Puedes tú, igual que Robin ganar el trofeo más preciado, como es la flecha de plata? Para Spectrum 48, 128 K. Commodore 64 ó 128 K. Amstrad CPC. P.V.P. **2.300**



The Arc Of Yesod: Nuevo Crash Mash. De Thor Computer Software. este juego es la continuación de Nades Of Yesod. Descubrirás un mundo galáctico verdaderamente sorprendente. Spectrum 48 ó 128 K. Commodore 64 ó 128 K. Próximamente Amstrad CPC. P.V.P. **2.300**



Marsport: de Gargoyle Game. Crash Mash. Este juego es la primera parte de la trilogía. «El asedio de la tierra». Las críticas de Microhobby «Marsport es un excelente juego sin igual». Amstrad CPC, Spectrum 48 ó 128 K. P.V.P. **2.300**



Four Pack: Amstrad CPC. De Software Proyects cuatro fantásticos juegos: Manic Miner, Binky, Jep Set Willy. Kars. Treasure Hunt. En disquette y ahora también en cassette, constituyendo una novedad sorprendente.
Disquette P.V.P. **4.000**
Cassette P.V.P. **2.300**



Sweevos World: de Gargoyle Game. Nuevo Hit. Es la más divertida aventura de dibujos animados en la que tú puedes participar. Para Amstrad CPC, Spectrum 48 ó 128 K. P.V.P. **1.900**

CREACION DE FUNCIONES DE USUARIO

- COMO DEFINIR TUS PROPIAS FUNCIONES
- USO DE LAS FUNCIONES DE ENTRADA
- FUNCIONES INTERESANTES

En BASIC no hay funciones para cosas como cubos, interés compuesto o letras mayúsculas. Pero en muchos ordenadores se pueden crear funciones especiales cuando se necesitan.

Tu ordenador es capaz de hacer muchas cosas —en especial las relacionadas con los cálculos matemáticos— con mucha más rapidez que tú. Pero no puede hacer más que algo que le indiques previamente: o lo dices tú tecleando un programa, o se lo dice un programa que cargues (con LOAD) en tu ordenador.

El lenguaje más común que la gente suele utilizar para programar es el BASIC, el que se inicializa automáticamente al encender el equipo. Tu ordenador dispone de varias funciones incorporadas en el BASIC estándar: son funciones como SIN, COS, SQR y otras muchas. Cada una de ellas está definida de tal forma que tu ordenador sabe cómo reconocer el comando y realizar la operación adecuada sobre un valor dado, calculando su seno, su coseno, su raíz cuadrada o lo que sea.

FUNCIONES EN BASIC

Todo esto está muy bien en la medida en que las funciones que quieras usar en tu programa sean también **palabras clave** del BASIC de tu ordenador. Pero hay muchas funciones comunes que no son parte del BASIC estándar; por ejemplo, no hay ninguna función que se llame CUBO, que te calcule automáticamente el valor de X*X*X. Si no está disponible una función que vayas a utilizar muy a menudo, tienes varias opciones. Puedes buscar la forma de evitar el uso de esa función en tu programa; también puedes añadir una subrutina que calcule la función deseada; y, lo más elegante de todo, puedes emplear la facilidad de las funciones definibles por el usuario que tiene tu ordenador. El **Commodore 64** y **Vic 20** disponene de esta facilidad, aunque es más bien limitada.

Literalmente las funciones definibles por el usuario te permiten configurar el BASIC a la medida de las necesidades de un programa particular. La forma básica del comando que define la función es la siguiente:

DEF FN a(x,y) = ... (lo que quieras que haga la función).

La letra entre paréntesis es el parámetro de la función. El **Commodore 64** y el **Vic 20** sólo pueden tener un parámetro por cada función.

Para ver cómo funciona el principio general en un programa real, teclea y ejecuta el siguiente ejemplo. En él se define la sencilla función mencionada anteriormente, es decir, te da el cubo de un número.

```
10 DEF FNC(X)=X*X*X
30 PRINT "NUMERO","CUBO"
40 FOR A=1 TO 13
50 PRINT A,FNC(A)
60 NEXT
```



LLAMADA A UNA FUNCION

Es muy fácil llamar a una función; para ello, cada vez que precises el resultado numérico del cálculo de dicha función, teclea la palabra clave FN y a continuación el nombre de la función deseada. Esto es lo que se hace en la línea 50 del programa. En este ejemplo el programa utiliza FNC(A), donde A es el número cuyo cubo has de calcular. Si quieres que tu programa utilice esta función, no tienes más que usar FNC(A) en todos los sitios en que en otro caso pondrías A*A*A.

Este mismo principio se puede aplicar para cualquier otro uso de funciones definidas por el usuario; de hecho, la llamada a funciones puede considerar como una forma de abreviatura en los cálculos.

Para una función tan sencilla como la de este ejemplo, podría pensarse que es más rápido utilizar la función de elevación a potencias que lleva incorporada el BASIC de tu ordenador. De hecho, incluso en este ejemplo, es realmente más rápido llamar a la función (o multiplicar el número varias veces por sí mismo) cuando la potencia es menor o igual que tres. Naturalmente, con DEF FN puedes abordar cálculos mucho más complejos que el de este ejemplo.

Aunque normalmente no interesará utilizar una función para este tipo tan sencillo de cálculos, te puede servir para familiarizarte con el uso de DEF FN en tu ordenador, experimentando con tus últimas versiones modificadas del último programa. Intenta modificarlas para definir nuevas funciones que dividan un número por 10, multipliquen por 2000 o calculen el seno. A continuación puedes utilizar las funciones matemáticas que tu ordenador ya tiene en su BASIC. Para comprobar si has definido una función correctamente, puedes compararla con lo que obtienes cuando el ordenador calcula la respuesta a un comando directo.

Durante los experimentos, es útil entender cómo funciona realmente DEF FN en tu ordenador.

NOMBRES DE FUNCIONES

Cada función que definas debe tener un nombre, utilizado para llamarla y emplearla en tus programas. El nombre está formado por una o dos letras situadas inmediatamente después de DEF FN. Por ejemplo: DEF FN ej (que empieza a definir una función llamada ej).

En realidad el nombre de la función en los **Commodore** puede contener más de dos letras o números, pero sólo se tendrán en cuenta los dos primeros. Además el primer carácter de cada nombre debe ser una letra. Por ejemplo, se admite un nombre tal como ACIDO, pero el ordenador no apreciaría diferencia entre esta función y una que se llamara ACTOR, o cualquier otro nombre que empezase por AC.

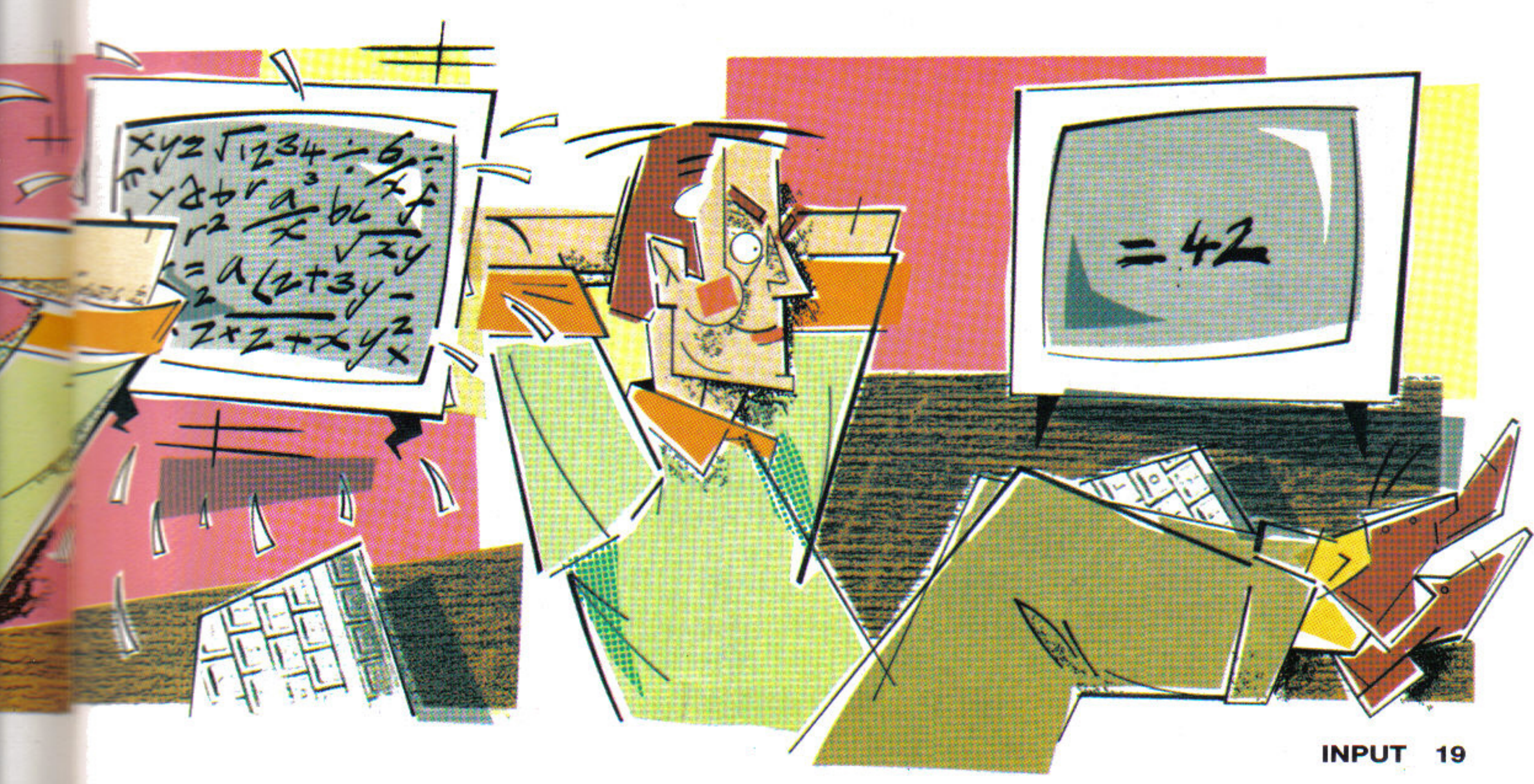
DEFINICION DE PARAMETROS

El siguiente paso para definir tu función es añadirle los «parámetros». Son los números o letras que figuran entre paréntesis después del nombre de la función.

Sirven para informar al ordenador de que la función utilizará esos números del programa en sus cálculos. En otras palabras, si empiezas definiendo tu función así:

DEF FN ej(a)

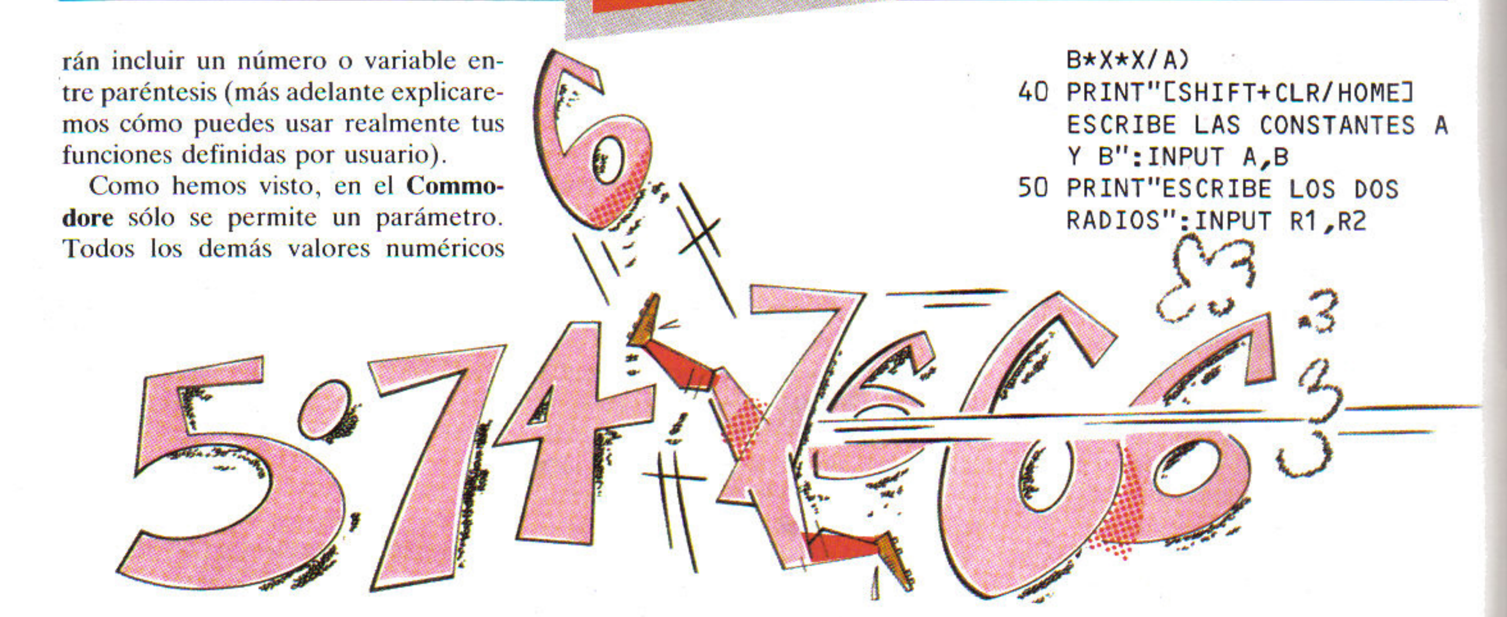
el ordenador esperará un número proporcionado por el programa. Así, cuando hagas una llamada para utilizar la función en tu programa, debe-



rán incluir un número o variable entre paréntesis (más adelante explicaremos cómo puedes usar realmente tus funciones definidas por usuario).

Como hemos visto, en el **Commodore** sólo se permite un parámetro. Todos los demás valores numéricos

```
   B*X*X/A)
40 PRINT"[SHIFT+CLR/HOME]
   ESCRIBE LAS CONSTANTES A
   Y B":INPUT A,B
50 PRINT"ESCRIBE LOS DOS
   RADIOS":INPUT R1,R2
```



que necesite la función han de ser valores numéricos reales que intervengan en la propia definición. Así, si quieres multiplicar un valor por 9.81 como parte de tu función, podrías utilizar un parámetro a para el valor que ha de ser cambiado por la función, y también usar 9.81 en la definición.

¿POR QUE USAR FUNCIONES?

¿Cuándo es mejor usar una función en una línea de programa en lugar de una instrucción directa?

La ventaja más clara de definir una función en vez de realizar una serie de cálculos, se presenta cuanto tienes que realizar el mismo cálculo varias veces. Con una función definida por el usuario puedes ahorrar memoria y tiempo.

Esto resulta útil con ecuaciones largas o funciones matemáticas que vayas a usar mucho en tus programas. Teclea y ejecuta el siguiente programa que DEFine una FUnciónN para calcular calcular valores con la ecuación de una elipse. Para el **Commodore 64** necesitarás el Simon's BASIC y para el **Vic** el cartucho Super Expander.

Al ejecutarlo tienes que introducir en los INPUT unos cuantos valores,

```
60 R=R1:Y1=FNY(0):R=R2:Y2=
   FNY(0)
70 IF Y2>Y1 THEN Y1=Y2
80 IF Y1>100 THEN Y1=Y2
90 HIRES 0,1
100 FOR K=-159 TO 159
110 IF B*R1*R1<B*K*K/A THEN
    130
120 R=R1:PLOT K+159,100+FNY
    (K),1:PLOT K+159,100-FNY
    (K),1
130 IF B*R2*R2<B*K*K/A THEN
    150
140 R=R2:PLOT K+159,100+
    FNY(K),1:PLOT K+159,100
    -FNY(K),1
```



después de lo cual el programa utiliza la función para dibujar dos elipses.

```
150 NEXT
160 GOTO 160
```

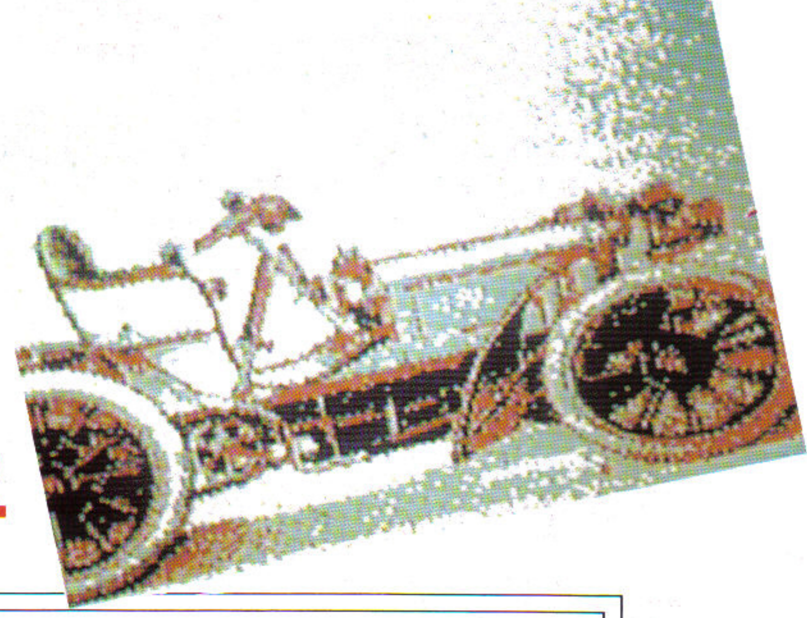
Teclea para Commodore-64

```
1 REM**UTILIZAR EL SIMON'S
  BASIC EN EL C-64
10 DEF FNY(X)=SQR(B*R*R-
```

Teclea para Vic-20

```
1 REM**UTILIZAR EL CARTUCHO
  DE AMPLIACION DEL BASIC
10 DEF FNY(X)=SQR(B*R*R-
```

CONCURSO DE ARTE POR ORDENADOR



El **Commodore 64** es uno de los microordenadores del momento con mejores capacidades para la producción de gráficos. Sin duda seréis muchos los que en algún momento habéis volcado los frutos de la inspiración sobre la pantalla, utilizando fórmulas matemáticas, dibujo *pixel* a *pixel* o alguna de las diferentes herramientas de dibujo existentes.
Ahora es el momento de enviar vuestras pantallas a **INPUT Commodore,** para ver premiada vuestra creatividad.
La única limitación impuesta para la recepción de los diseños es que deben llegarnos sobre alguno de los habituales soportes informáticos, volcado de impresora, cinta, *diskette,* etc. Pueden venir acompañados de una fotografía, pero esta no es necesaria ni suficiente.
La fecha límite de recepción de las pantallas es el 30 de marzo. La decisión de nuestro jurado será inapelable.
En el número de mayo se publicarán las pantallas premiadas, quedando los derechos de copia propiedad de **INPUT Commodore.**
La presentación de trabajos implica aceptar las anteriores condiciones.



PREMIOS: microbyte

2 PRIMEROS PREMIOS:

1 tableta Koala Pad y los 11 programas de nuestro catálogo Commodore:

—Fighter Pilot
—Cave Fighter
—Anticipital
—Cliff Hanger
—Gryphon
—Ant Attact
—Slap Shot
—Zaga Mision
—Pole Position
—Black Thunder

y el programa del año en Inglaterra, Mini office (4 programas en uno): Base de Datos; Hoja de cálculo; Procesador de textos y gráficos.

2 SEGUNDOS PREMIOS:

1 Tableta Koala Pad
4 Programas Commodore:

—Zaga Mission
—Cava Fighter
—Cliff Hanger

y el fabuloso mini Office

2 TERCEROS PREMIOS

Los 11 programas de nuestro catálogo Commodore.



```
    B*X*X/A)
40 PRINT"[SHIFT+CLR/HOME]
    ESCRIBE LAS CONSTANTES A
    Y B":INPUT A,B
50 PRINT"ESCRIBE LOS DOS
    RADIOS":INPUT R1,R2
60 R=R1:Y1=FNY(0):R=R2:Y2=
    FNY(0)
70 IF Y2>Y1 THEN Y1=Y2
80 IF Y1>511 THEN Y1=Y2
90 GRAPHIC 2
100 FOR K=-511 TO 511
110 IF B*R1*R1<B*K*K/A THEN
     130
120 R=R1:PLOT K+511,511+FNY
     (K),1:PLOT K+511,511-FNY
     (K),1
130 IF B*R2*R2<B*K*K/A THEN
     150
140 R=R2:PLOT K+159,100+
     FNY(K),1:PLOT K+159,100
     -FNY(K),1
150 NEXT
160 GOTO 160
```

El programa utiliza los familiares gráficos del **Commodore** para dibujar las elipses, usando la función definida en la línea 10 para evaluar las coordenadas de cada elipse. Observa que en la definición de una función se puede utilizar otra de función de las incorporadas en el ordenador, en este caso, la función SQR.

DALE MAS INTERES

Aquí tienes otro ejemplo de funcionamiento de una sentencia DEF FN, en forma de programa de utilidad que, aunque corto, puede ser de gran ayuda en el cálculo de los intereses que podrían producir tus ahorros.

El programa define una función que calcula la cantidad de dinero que tendrías después de transcurrir un tiempo determinado, y para un determinado tipo de interés. Tienes que introducir en los INPUTs la cantidad de dinero con la que empiezas, la tasa o tanto por ciento de interés y el período de tiempo.

```
10 DEF FN C(T)=INT(AM*((1+R/
    100)^T)*100)/100
20 INPUT "CUANTO ";AM
30 INPUT "TASA DE INTERES
    (%)"; R
40 INPUT "NO. DE UNIDADES DE
    TIEMPO ";T
50 PRINT "CANTIDAD TOTAL
    DESPUES DE INTERESES=";
    FNC(T)
60 PRINT:PRINT"PRESIONA
    CUALQUIER TECLA PARA OTRO
    CALCULO NUEVO"
70 GET A$:IF A$=""THEN 70
80 GOTO 20
```

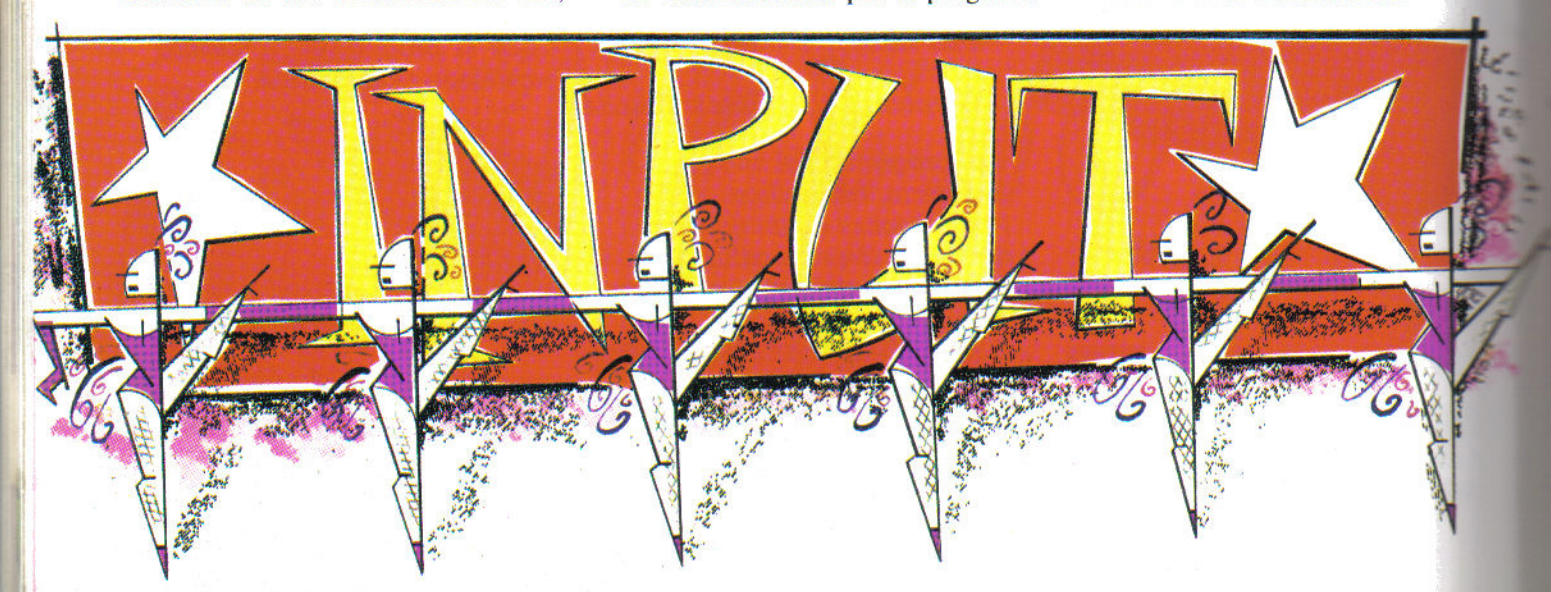
El funcionamiento de este programa es muy sencillo; define la función y pasa a las sentencias INPUT por medio de las cuales puedes introducir todos los detalles relativos a tus ahorros. Las variables utilizadas por el programa son T para el período de tiempo; C para el capital o cantidad de partida y R para el rédito o tipo de interés.

El valor calculado por el programa es el del interés compuesto, es decir, el interés producido en un período de tiempo se acumula al interés de los períodos anteriores. Esto es tenido en cuenta automáticamente por la función. Si, por ejemplo, la tasa de interés es de 10%, después de un período de tiempo tendrás la suma original más el 10% de la misma, o lo que es lo mismo, la suma original más 0,1 veces esa suma. Esta expresión aparece en la función como R/100+1. Así, en este caso el resultado después de transcurrir el primer período es 1.1 veces la cantidad con la que empezaste. Después de dos períodos el resultado será 1.1 veces la nueva suma de partida, que a su vez ya era 1.1 veces la suma original. Es decir, es 1,1*1,1 veces la cantidad con la que empezaste. Y 1,1*1,1 es 1,1 ↑ T. Esta es la segunda parte de la función, cuya forma completa, tal como aparece en el programa es (R/100+1) ↑ T. La parte restante de la expresión multiplica esto por la cantidad original C y redondea el resultado con dos cifras significativas, para lo cual multiplica por 100, utiliza la función INT y divide de nuevo por 100.

Puedes utilizar las dos fórmulas que siguen para crear dos funciones definidas por el usuario que no tiene tu ordenador. Son las fórmulas de ARC SIN y ARC COS:

```
ASN(X)=ATN(X/SQR(-X*X+1)
ACS(X)=-ATN(X/SQR(-X*X+1))
        +1.5708
```

También puedes usar fórmulas para otras funciones matemáticas.



CRONOMETRO, PROGRAMA DE UTILIDAD

El objetivo del programa es convertir al Commodore-64 **en un verdadero cronómetro, capaz de medir, además del tiempo total, una cantidad de hasta cinco tiempos intermedios.**

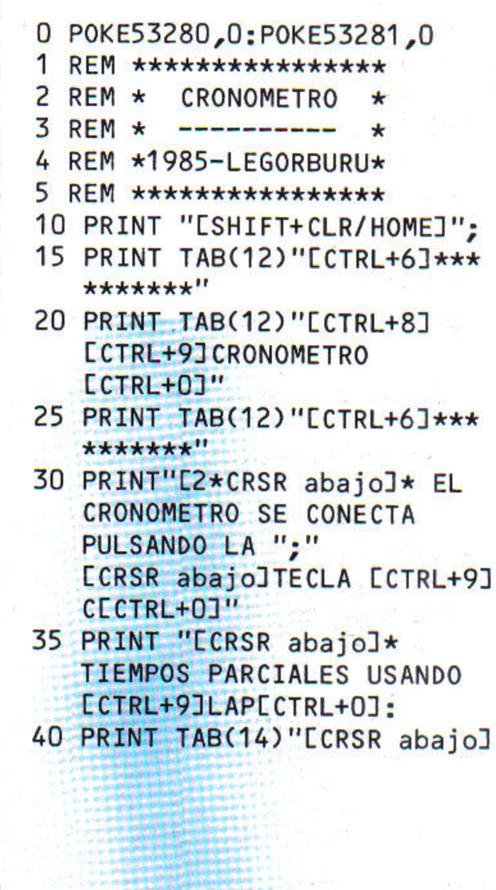
El programa se fundamenta en la utilización de la variable TI, que corresponde al cronómetro interno incorporado en el ordenador.

Hay dos partes bien diferenciadas en el programa:

La primera, donde se encuentra la presentación, con las instrucciones a seguir para la utilización del programa.

En la segunda está el sistema operativo del programa, empieza en la línea 100, que es donde se pone en marcha el cronómetro mediante una instrucción GET. Entre las líneas 135 a 170 tenemos las claves, también mediante instrucciones GET, para las medidas de tiempo parciales y el tiempo total. En la línea 105 es donde se toma la medida inicial en el contador del cronómetro interno, para obtener los tiempos intermedios y el tiempo final se realiza la correspondiente lectura del contador y se halla su diferencia con la medida inicial, por último, para imprimir el tiempo, sólo es necesario pasarlo a segundos.

Ignacio Javier Legorburu.



```
0 POKE53280,0:POKE53281,0
1 REM ****************
2 REM *  CRONOMETRO  *
3 REM *  ----------  *
4 REM *1985-LEGORBURU*
5 REM ****************
10 PRINT "[SHIFT+CLR/HOME]";
15 PRINT TAB(12)"[CTRL+6]***
   *******"
20 PRINT TAB(12)"[CTRL+8]
   [CTRL+9]CRONOMETRO
   [CTRL+0]"
25 PRINT TAB(12)"[CTRL+6]***
   *******"
30 PRINT"[2*CRSR abajo]* EL
   CRONOMETRO SE CONECTA
   PULSANDO LA ";"
   [CRSR abajo]TECLA [CTRL+9]
   C[CTRL+0]"
35 PRINT "[CRSR abajo]*
   TIEMPOS PARCIALES USANDO
   [CTRL+9]LAP[CTRL+0]:
40 PRINT TAB(14)"[CRSR abajo]
```

```
    LAP1-[CTRL+9]Q[CTRL+0]"
45 PRINT TAB(14)"LAP2-
    [CTRL+9]W[CTRL+0]"
50 PRINT TAB(14)"LAP3-
    [CTRL+9]E[CTRL+0]"
55 PRINT TAB(14)"LAP4-
    [CTRL+9]R[CTRL+0]"
60 PRINT TAB(14)"LAP5-
    [CTRL+9]T[CTRL+9]"
65 PRINT "[CRSR abajo]*
    TIEMPO TOTAL PULSANDO LA
    TECLA [CTRL+9]M[CTRL+0]"
100 GET C$:IF C$<>"C"
     THEN 100
105 C=TI
110 PRINT"[SHIFT+CLR/HOME]";
115 PRINT TAB(12)"[CTRL+6]**
     ********"
120 PRINT TAB(12)"[CTRL+8]
     [CTRL+9]CRONOMETRO
     [CTRL+0]"
125 PRINT TAB(12)"[CTRL+6]**
     ********"
130 PRINT "[2*CRSR abajo]
     [CTRL+9]c[CTRL+0]"
135 GET T$
140 IF T$="Q"THEN 175
145 IF T$="W"THEN 195
150 IF T$="E"THEN 215
155 IF T$="R"THEN 235
160 IF T$="T"THEN 255
165 IF T$="M"THEN 275
170 GOTO 135
175 Q=TI
180 QQ=Q-C
185 PRINT"[CLR/HOME]":PRINT
     TAB(8)"[7*CRSR abajo]
     [CTRL+9]LAP1[CTRL+0]=";
     QQ/60;"SGS."
190 GOTO 135
195 W=TI
200 WW=W-C
205 PRINT"[CLR/HOME]":PRINT
     TAB(8)"[9*CRSR abajo]
     [CTRL+9]LAP2[CTRL+0]=";
     WW/60;"SGS."
210 GOTO 135
215 E=TI
220 EE=E-C
225 PRINT"[CLR/HOME]":PRINT
     TAB(8)"[11*CRSR abajo]
     [CTRL+9]LAP3[CTRL+0]=";
     EE/60;"SGS."
230 GOTO 135
235 R=TI
240 RR=R-C
245 PRINT"[CLR/HOME]":PRINT
     TAB(8)"[13*CRSR abajo]
     [CTRL+9]LAP4[CTRL+0]=";
     RR/60;"SGS."
250 GOTO 135
255 T=TI
260 TT=T-C
265 PRINT"[CLR/HOME]":PRINT
     TAB(8)"[15*CRSR abajo]
     [CTRL+9]LAP5[CTRL+0]=";
     TT/60;"SGS."
270 GOTO 135
275 M=TI
280 MM=M-C
285 PRINT"[CLR/HOME]":PRINT
     "[19*CRSR abajo][CTRL+9]
     TIEMPO TOTAL[CTRL+0]=";
     MM/60;"SGS."
290 GOTO 290
```

ESTRUCTURA TUS PROGRAMAS

- ¿QUE ES LA PROGRAMACION ESTRUCTURADA?
- USO DE DIAGRAMAS DE FLUJO
- COMO USAR ESTRUCTURAS EN LOS PROGRAMAS BASIC

Un buen diseño hará que tus programas sean más fáciles de entender y que funcionen mejor. También puede marcar la diferencia entre que se ejecuten eficientemente o fallen estrepitosamente.

Cuando decides escribir tu primer programa para ordenador, lo normal es que sientas una incontrolable urgencia de sentarte ante el teclado y empezar a teclear inmediatamente alguna parte del programa.

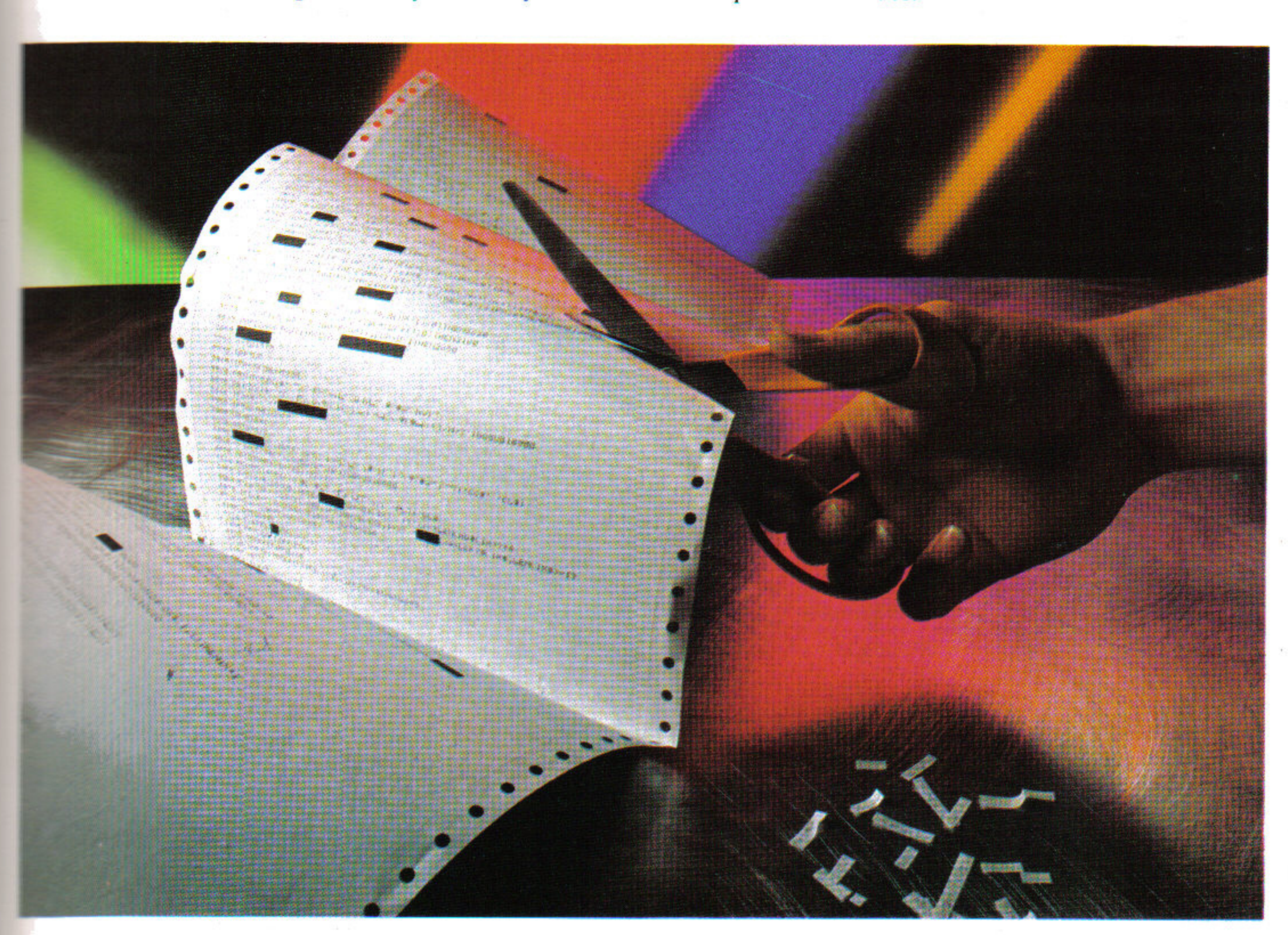
Puede ser que las primeras líneas que escribas funcionen bien. Te sentirás muy satisfecho contigo mismo y añadirás unas cuantas líneas más. Conseguirás que éstas funcionen también y seguirás añadiendo líneas aquí y allá, probando a medida que avanzas, hasta que le programa se extienda, probablemente a unos cientos de líneas.

Entonces, de repente, se produce el desastre. Añades unas líneas de programa y ya no funciona nada. No puedes encontrar ninguna razón por la que haya podido dejar de funcionar. Cambias esta línea, aquella otra, pero no hay nada que hacer.

Sin embargo, no te des por vencido. Hay maneras de hacer que esto no suceda, y lo único que face falta es un poco de organización. Si sigues unas cuantas sencillas reglas cuando escribas tus programas, no tendrás el menor problema.

PROGRAMACION ESTRUCTURADA

El BASIC estándar fue diseñado para que fuera fácil de utilizar, pero tiene muy poca **estructura** comparado con otros lenguajes, lo cual hace más difícil escribir programas estructurados.



Las estructuras son los bloques básicos que utilizas para construir la forma general de tus programas. En el BASIC son IF... THEN, FOR... NEXT, GOTO y GOSUB. Ya sabes cómo se utiliza cada una de ellas por separado, pero nuestro objetivo actual es ponerlas juntas de una forma ordenada y legible.

Es muy fácil y rápido escribir unas cuantas líneas de programa que funcionen a la primera (o casi a la primera) y sean relativamente sencillas de entender por cualquiera. De hecho, si el programa es corto, no hace falta tomar especiales precauciones para estructurarlo. El problema se presenta cuando quieres escribir un programa largo que haga algo útil. En este caso, si empiezas escribiendo un fragmento de programa y le vas añadiendo más y más trozos, lo más probable es que termines irremediablemente perdido, a menos que tengas una memoria de elefante, naturalmente.

Lo que tienes que hacer es sentarte y diseñar el programa sistemáticamente. Casi todos los programas empiezan con una vaga idea de algo que tú quieres que haga el ordenador, y cuanto más complejo sea ese «algo» más vagas serán tus ideas. El concepto básico puede corresponder a un juego o un paquete de proceso de textos, o cualquier otra cosa tan amplia que te sea imposible mantener todas las ideas en la cabeza al mismo tiempo.

Para empezar sitúate en algún lugar lejos del ordenador, preferiblemente en otra habitación, para evitar tentaciones, y determina y escribe de una forma general lo que quieres hacer. Este es el principio de lo que se llama una **especificación de diseño.** Por ejemplo, podrías escribir:

Sistema de índices.

El programa permitirá crear, actualizar, borrar, ordenar y listar registros en memoria. Se accederá a los registros usando una palabra clave. Todo el conjunto de registros puede ser almacenado (SAVE) en un fichero y cargado (LOAD) desde el mismo.

A continuación tienes que dividir esta descripción tan general en unos cuantos pasos lógicos o **módulos.** Las operaciones que intervienen en cada módulo probablemente serán aún muy complicadas, por lo que también deberán ser fraccionadas en secciones más pequeñas, hasta que creas que los módulos de más bajo nivel son los suficientemente cortos para ser manejados. La figura 1 muestra cómo podría hacerse esto para el Sistema de Indices.

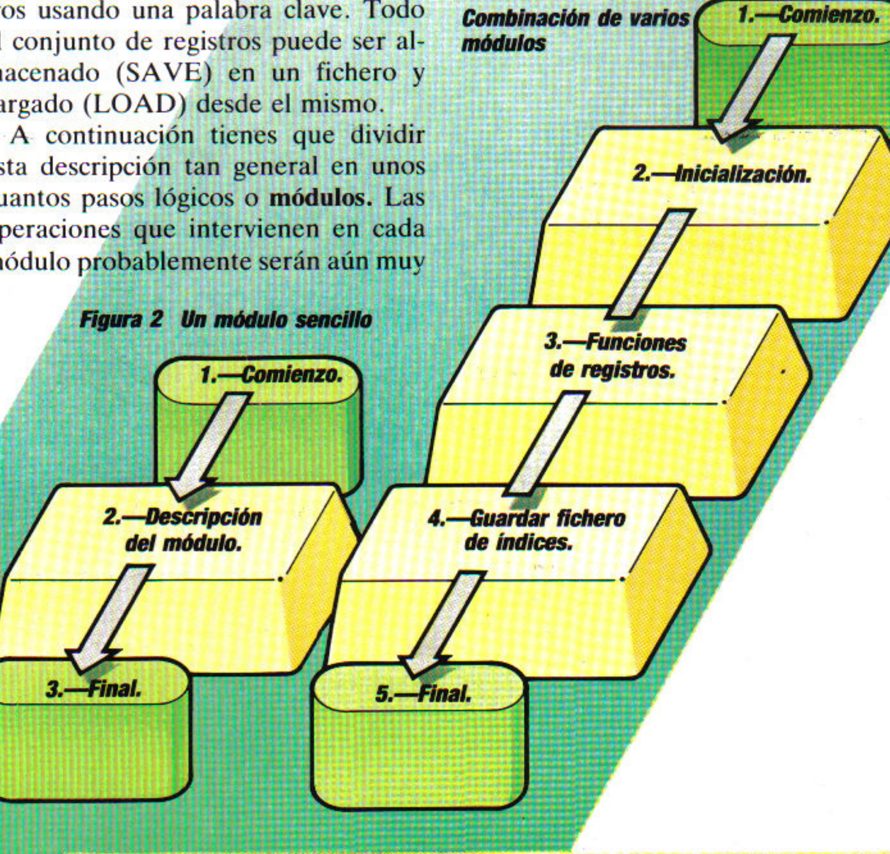


Figura 2 Un módulo sencillo

Figura 3 Combinación de varios módulos

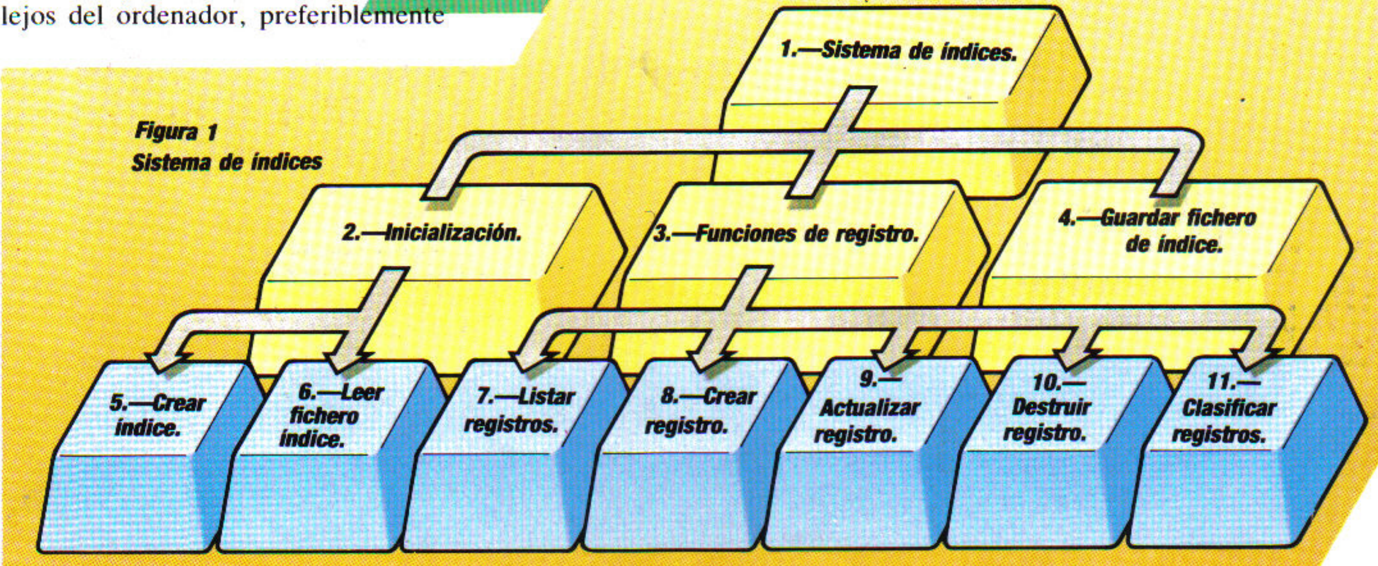


Figura 1 Sistema de índices

Cada una de las secciones más pequeñas no deberían superar la extensión de una página —unas 60 líneas— pero si la longitud es la mitad, mejor todavía. Cada uno de estos módulos de bajo nivel debe ser muy fácil de comprender. Eventualmente, cada módulo terminará como una subrutina de tu programa.

El proceso de fragmentar los problemas se irá haciendo más fácil con la experiencia. Y, como siempre, la mejor manera de aprender a hacerlo es intentarlo. Este método de dividir el problema se conoce como diseño de arriba a abajo. Empezaste por lo más alto (la descripción general del programa) y seguiste procediendo hasta lo más bajo (los módulos de más bajo nivel). Sin embargo, todavía no has decidido el orden de los módulos, es decir, el orden en que son ejecutados por el programa. Es lo que vamos a ver a continuación.

DIAGRAMAS DE FLUJO

Una manera de especificar el orden de los módulos consiste en utilizar **diagramas de flujo.** Son fáciles de usar y al ser esquemáticos, son fáciles de entender y seguir de una ojeada. Igual que el BASIC, no son estructurados en sí mismos, por lo que deben ser utilizados con precaución. Te puedes encontrar enredado y confundido con un diagrama de flujo, con la misma facilidad con que te ves atrapado en un enmarañado programa. Sin embargo, siguiendo unas cuantas reglas sencillas, los gráficos te pueden servir para ordenar un programa de una forma directa y estructurada.

La figura 2 muestra cómo puede ser especificado un módulo sencillo. El programa va siguiendo las líneas en dirección de las flechas y los bloques describen lo que sucede en cada etapa. Para ejecutar una serie de módulos en sucesión, no tienes más que añadir más bloques en el orden correcto entre Comienzo y Final (figura 3).

Un diagrama de flujo sencillo como éste vale para un programa en el que no se toman decisiones. No obstante, la potencia del ordenador se debe en gran medida a su capacidad de tomar decisiones dentro de un programa. Para esto es la familiar instrucción IF ... THEN que es una de las estructuras más usadas en BASIC.

Echemos un vistazo a todas las estructuras diferentes y a cómo pueden representarse con diagramas de flujo.

IF ... THEN ... ELSE

Aunque hay diferencias en la forma en que los ordenadores utilizan este comando, el conjunto IF ... THEN ... ELSE es la base de todas las decisiones que el ordenador tiene que tomar. Su diagrama de flujo se muestra en la figura 4 y en BASIC se codifica así:

```
100 IF condicion THEN
    sentencia 1 ELSE
    sentencia 2
```

El significado es: si la condición es verdadera, se ejecuta la sentencia 1, en caso contrario se ejecuta la sentencia 2.

Como puedes ver, el diagrama de flujo no tiene más que un punto de en-

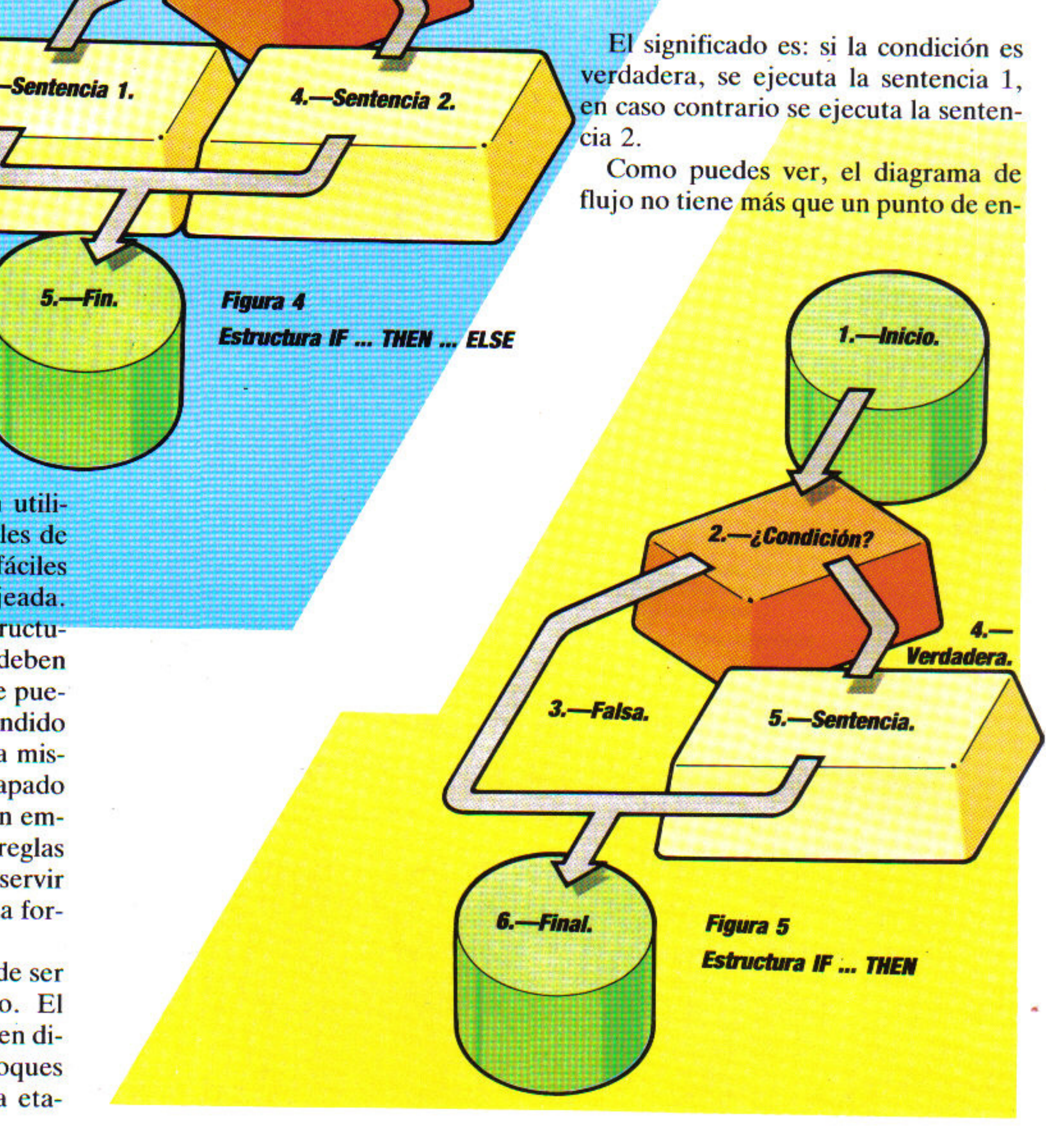


Figura 4
Estructura IF ... THEN ... ELSE

Figura 5
Estructura IF ... THEN

trada y uno de salida. Esto facilita mucho la comprobación y la depuración de errores, pues conoces exactamente dónde comienza y acaba dicha sección del programa. Esta es de hecho una regla muy importante de la programación estructurada: para cada sección de programa debe haber sólo una entrada y una salida.

Casi ninguna versión de BASIC dispone la parte ELSE de la sentencia. En tu ordenador no está disponible. Pero no importa demasiado ya que puedes simularla usando el GOTO así:

```
100 ...
110 IF condicion THEN GOTO
    140
120 sentencia 2:REM esta es
    la componente ELSE
130 GOTO 150
140 sentencia 1:REM esta es
    la componente ELSE
150 ...
```

Naturalmente, los números de las líneas no tienen por qué ser los que hemos puesto aquí. Además en THEN y ELSE, se puede incluir más de una sentencia. Por ejemplo, aquí tienes una sección de programa para ordenar correctamente dos números. Constituye la base de una rutina de clasificación alfabética que veremos en otros artículos más adelante. En este caso, la componente ELSE contiene cuatro sentencias:

```
100 IF primero<=segundo THEN
    GOTO 160
110 LET temporal=primero
120 LET primero=segundo
130 LET segundo=temporal
140 LET orden$="erroneo"
150 GOTO 170
160 LET orden$="correcto"
```

Sería equivalente escribir este ejemplo con ELSE, pero en tal caso habría que escribirlo todo en una línea. Se pueden utilizar sentencias múltiples siempre que estén separadas por dos puntos.

```
100 IF primero>segundo THEN
    temporal=primero:
    primero=segundo:
    segundo=temporal:
    orden$="erroneo"ELSE
    orden$="correcto"
```

Como ves, no resulta fácil leer o entender programas escritos usando sentencias largas. Por ello debes intentar evitarlas todo lo posible.

Por último, en muchos casos puede que no necesites para nada la componente ELSE. El correspondiente diagrama de flujo es el de la figura 5, cuya codificación es:

```
100 IF condicion THEN
    sentencia
```

que, naturalmente es la sentencia IF ... THEN.

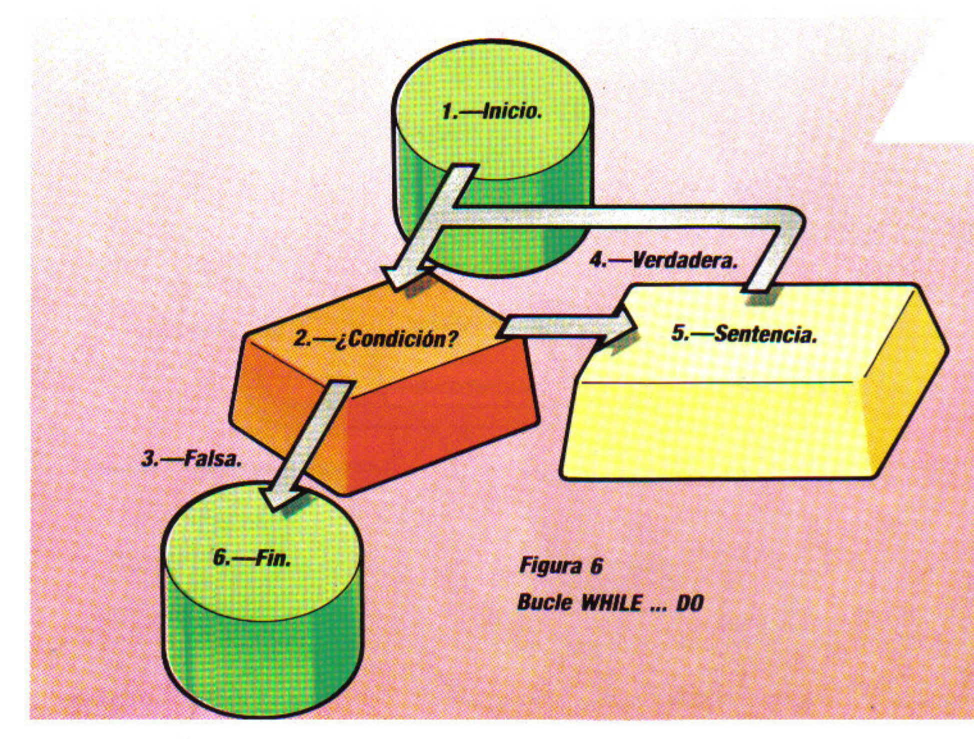


Figura 6
Bucle WHILE ... DO

ESTRUCTURAS ANIDADAS

Las líneas con IF ... THEN ... ELSE pueden ir anidadas. Esto quiere decir que una o ambas sentencias entre las que puede escoger la sentencia con IF ... THEN, puede a su vez ser un IF ... THEN. Por ejemplo, la siguiente sección de programa lleva la cuenta de cuántos juegos van ganando dos jugadores y presenta en pantalla el resultado de cada juego:

```
100 IF T1<>T2 THEN GOTO 130
110 PRINT "Tirada"
120 GOTO 190
130 IF T1<T2 THEN GOTO 170
140 PRINT "Gana el primer
    jugador"
150 LET J1=J1+1
160 GOTO 190
170 PRINT "Gana el segundo
    jugador"
180 LET J2=J2+1
190 ...
```

Todas las estructuras pueden anidarse en cualquier combinación, y en teoría con cualquier profundidad. Sin embargo, cuanto más anidados pongas, menos legible será el programa; un límite práctico son tres a lo sumo cuatro estructuras anidadas. Si te encuentras con que te hacen falta más, divide el programa en módulos más pequeños o subrutinas.

Fíjate otra vez en el último programa. Es realmente difícil seguir lo que está haciendo, a pesar de que está perfectamente bien estructurado. Una forma eficaz de hacer más legibles las sentencias anidadas, es retranquear o escalonar (indentar) las líneas de programa. Esto es posible reescribiendo el último programa así:

PERSONAL PRINTER

RITEMAN C+

BUILT-IN COMMODORE INTERFACE

MANUAL DE INSTRUCCIONES EN CASTELLANO

NOVEDAD: C + versión NLQ

INFORUNNER

RITEMAN C+

ESPECIFICACIONES

1. **Método de impresión**
   Serie de impactos por matriz de 9×9
2. **Cabezal impresor**
   9 agujas (reemplazable) con expectativa de vida de 50 millones de caracteres a 100 % de uso contínuo.
3. **Velocidad de impresión**
   Impresión a 105 cps uni o bi-direccional, búsqueda lógica optimizada, capacidad de impresión 45 lpm a 10 cpi., 200 ms de salto de línea con espaciado de 1/6".
4. **Espaciado interlínea**
   1/6", 1/8", 7/72", programable N/72", N/216".
5. **Alimentación de papel**
   a) Tractor ajustable entre 4" y 10".
   b) Alimentación a fricción para hojas sueltas entre 4" y 9,5".
6. **Corte de papel**
   Se puede imprimir desde principio de página hasta una pulgada de fin de papel.
7. **Selección de caracteres**
   96 caracteres ASCII, 96 itálicos, 82 caracteres gráficos Commodore. Definición de caracteres por usuario a través de software fácilmente.
8. **Gráficos**
   Normal 480 puntos por línea. Doble densidad 960.
9. **Modos de impresión**
   Modo Commodore
   Standard, reverse (negativo), mayúsculas y minúsculas y caracteres gráficos.
   Modo plus
   Standard, doble impresión, enfatizado, itálica, supra y subíndices mitad altura.
10. **Tamaños de impresión**
    Modo Commodore
    a) Normal
    b) Expandido
    c) Expandido doble
    d) Expandido triple
    e) Expandido cuádruple.
    Modo plus
    a) Normal (10 cpi), 2,1 mm (a)×2,55 mm(h).
    b) Normal expandido (5 cpi)
    c) Comprimido (17 cpi)
    d) Comprimido-expandido (8,5 cpi).
11. **Test escritura**
    Manteniendo apretado LF (salto de línea) cuando se acciona el interruptor de puesta en marcha ON.
12. **Subrayado contínuo.**
13. **Tabulador**
    Existen 32 posiciones de tabulación por línea.
14. **Longitud de formato**
    El standard de 66 líneas, con espaciado de líneas y longitud del formato programable.
    VFU eléctrico.
15. **Tamaño buffer**
    1 línea standard.
16. **Diagnóstico de detección de errores en modo Commodore.**
17. **Posicionador de papel contínuo**
    Desplazable parte anterior o posterior.
18. **Interface**
    Directo a Commodore salida RS232 a través de conector DIN (el cable se suministra con la impresora).
19. **Grosor papel**
    Máximo 2 copias, 40 g/m² peso contínuo, sensor papel.
20. **Operación con interruptores**
    De fácil acceso.
21. **Caracteres por línea**
    Normal: 80
    Normal expandido: 40
    Comprimido: 132
    Comprimido expandido: 66
22. **Cinta impresora**
    Autorretintada, mediante tampón interno, cartucho compacto de cinta negra fácilmente reemplazable y bajo coste.
23. **Tamaño**
    115 (h) × 275 (f) × 405 (a).
24. **Tensión alimentación**
    220 v/ 50 Hz.
25. **Entrada de papel**
    Frontal-horizontal.
26. **Retroceso**
    Por software.
27. **Instrucciones de formateado de tablas alfanuméricas**
28. **Elevadores inferiores standard**
    (para ubicación del papel contínuo debajo de la impresora).
29. **Motor: paso a paso**
30. **Indicadores:**
    • Conexión general
    • Fin de papel
    • On line
31. **Interruptores**
    ON/OFF, On line (SEL), salto de página (FF), salto de línea (LF).
32. **Compatible con la mayor parte de software existente**
    (incluso Commodore).

TODAS LAS ESPECIFICACIONES PUEDEN SER CAMBIADAS SIN PREVIO AVISO. * Commodore es marca registrada de Commodore Bussines Machine, Inc.

```
100 IF T1<>T2 THEN GOTO 130
110  PRINT "Tirada"
120 GOTO 190
130 IF T1<T2 THEN GOTO 170
140  PRINT "Gana el primer
     jugador"
150  LET J1=J1+1
160 GOTO 190
170  PRINT "Gana el segundo
     jugador"
180  LET J2=J2+1
190 ...
```

Otra forma de hacer que la estructura de tu programa sea más clara es insertar líneas en blanco, separando las diferentes secciones y utilizar sentencias REM.

WHILE ... DO

La otra estructura esencial es la sentencia WHILE ... DO. Permite la existencia de bucles en un programa y es una de las mejores formas de crearlos. Hace que el ordenador haga (DO) algo una y otra vez mientras (WHILE) sea verdadera una cierta condición. No te preocupes si tu ordenador no tiene en su repertorio las palabras DO y WHILE. No están disponibles en casi ninguna formas de BASIC. Pero puedes construir una estructura que haga lo mismo, utilizando IF ... THEN y GOTO.

En la figura 6 se presenta el diagrama de flujo, y la codificación en BASIC es:

```
100 ...
110 IF NOT(condicion) THEN
    GOTO 140
120 sentencia
130 GOTO 110
140 ...
```

Observa que la línea 110 figura IF NOT (condición)...; esto significa que se comprueba si la condición es falsa, al contrario de lo normal. Pero no hay problema. Si la condición es A = B, NOT(A = B) será entonces A<>B. Análogamente, NOT(A<B) será A>=B, etc. Realmente puedes escribir NOT (A=B) si te gusta y el ordenador entiende lo que quieres decir.

P y R

¿Tiene algún significado especial la forma de los bloques o cajas utilizados en los gráficos de flujo?

Sí, los diagramas de flujo se dibujan utilizando símbolos normalizados. Hay cinco formas principales:

1.º Los bloques redondeados se llaman bloques **terminales** y muestran dónde empieza y termina un programa.

2.º Los círculos son símbolos **conectores,** utilizados al principio y al final de un módulo.

3.º Los rectángulos son bloques de **instrucción** y contienen sentencias de programa.

4.º Los rombos son bloques de **decisión.** Siempre hay al menos dos caminos de salida de estos bloques, dependientes de una decisión que se toma dentro del mismo.

5.º Finalmente, los paralelogramos (no mostrados en este artículo) son bloques de **entrada/salida,** que indican cualquier información que entra al programa, y cualquier salida por pantalla o impresora.

Aquí tienes como ejemplo un corto programa que utiliza un bucle WHILE y que puede servirte para contar el tiempo al hacer un nuevo pasado por agua:

Teclea para Commodore-64

```
5 PRINT"[SHIFT+CLR/HOME]
  [CTRL+1]":POKE 53281,4
10 PRINT TAB(16)"[CTRL abajo]
   TEMPORIZADOR"
20 INPUT"[3*CRSR abajo]
   [5*ESPACIO]CUANTOS
   MINUTOS PRECISAS";T
30 PRINT TAB(5)"[3*CRSR
   abajo]PRESIONA LA BARRA
   DE ESPACIOS PARA
   COMENZAR"
40 GET A$:IF A$="" THEN GOTO
   40
50 PRINT"[SHIFT +CLR/HOME]"
60 PRINT TAB(15)
   "TEMPORIZANDO"
70 TI$="000000"
75 REM COMIENZO DEL BUCLE
   WHILE
80 IF VAL(TI$)=>T*100 THEN
   GOTO 110
90 PRINT"[CLR/HOME][5*CRSR
   abajo]"TAB(15);RIGHT$
   (TI$,2)" SEGUNDOS"
100 GOTO 80
105 REM FIN DEL BUCLE WHILE
110 PRINT TAB(16)"[CRSR
    abajo]HECHO!"
120 POKE 54296,15:POKE
    54278,128:POKE 54276,17:
    POKE54273,50
130 FOR D=1 TO 200:NEXT:POKE
    54276,0:POKE 54278,0
```

Teclea para Vic-20

```
5 PRINT"[SHIFT+CLR/HOME]
  [CTRL+1]":POKE 36879,29
10 PRINT TAB(16)"[CTRL abajo]
   TEMPORIZADOR"
20 INPUT"[3*CRSR abajo]
   [5*ESPACIO]CUANTOS
   MINUTOS PRECISAS";T
30 PRINT TAB(5)"[3*CRSR
   abajo]PRESIONA LA BARRA
   DE ESPACIOS PARA
   COMENZAR"
40 GET A$:IF A$="" THEN GOTO
   40
50 PRINT"[SHIFT +CLR/HOME]"
60 PRINT TAB(15)
   "TEMPORIZANDO"
70 TI$="000000"
75 REM COMIENZO DEL BUCLE
   WHILE
80 IF VAL(TI$)=>T*100 THEN
   GOTO 110
90 PRINT"[CLR/HOME][5*CRSR
   abajo]"TAB(4);RIGHT$
   (TI$,2)" SEGUNDOS"
100 GOTO 80
105 REM FIN DEL BUCLE WHILE
110 PRINT TAB(16)"[CRSR
    abajo]HECHO!"
120 POKE 36878,15:POKE
    36876,200
130 FOR D=1 TO 200:NEXT:POKE
    36876,0
```

liminar, más otros que te permitan enlazar el juego.

Al dibujar este mapa, acuérdate de marcar los sentidos en que se puede circular en cada habitación, porque puede ser que haya puertas que quieras que sólo puedan cruzarse en una dirección, acompañadas de un mensaje tal como éste:

LA PUERTA SE CIERRA DE GOLPE JUSTO DETRAS DE TI

Las líneas de trazos que hay junto al cuarto oscuro indican que el aventurero sólo podrá ir en esa dirección si se cumplen ciertas condiciones. En este caso, la condición es que el aventurero tenga la lámpara y la haya encendido para poder ver las salidas.

Es muy difícil predecir cuántos lugares se pueden meter en una cantidad dada de RAM. La dificultad surge porque hay muchas cosas que se disputan el espacio de memoria en el programa de la aventura: descripciones de lugares, palabras que quieres que el programa reconozca, número de objetos y lo que quieres que se haga con ellos, número de enigmas y su complejidad, etc.

Cuando hayas escrito unas cuantas aventuras pequeñas y comprobado cuánta memoria ocupan, tendrás una idea de lo que puedes meter en tu máquina.

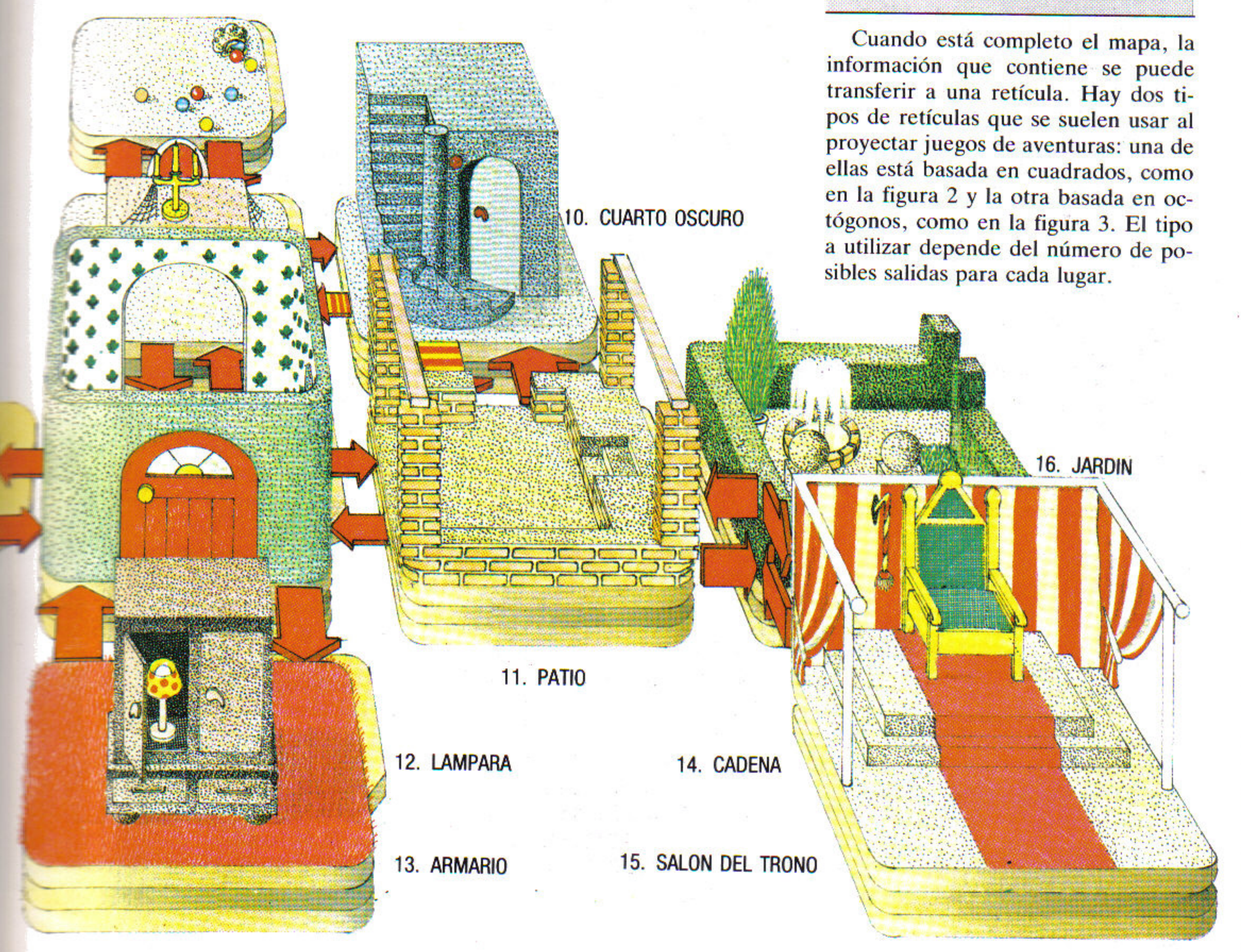
Los poseedores de un ordenador de 16 K pronto descubrirán que es casi imposible escribir una aventura en gran escala en una cantidad tan pequeña de RAM. Sin embargo, la aventura que desarrollaremos en los siguientes capítulos sólo tiene unos pocos lugares —12 en total— por lo que es suficientemente pequeña para no causarte dolores de cabeza.

El mapa para la búsqueda de la joya del globo ocular podría ser algo parecido a la figura 1. En esta fase las descripciones de los lugares han de mantenerse cortas. Se han dibujado los enlaces y se ha decidido el punto de partida. Esto es muy importante, ya que afecta a la manera en que se aborda la aventura, al orden en que aparecen los objetos y se plantean los *puzzles*.

También están marcados los objetos en sus lugares. Los objetos que aparecerán más tarde, por ejemplo la joya, es mejor listarlos a un lado del mapa.

PASO DEL MAPA A LA RETICULA

Cuando está completo el mapa, la información que contiene se puede transferir a una retícula. Hay dos tipos de retículas que se suelen usar al proyectar juegos de aventuras: una de ellas está basada en cuadrados, como en la figura 2 y la otra basada en octógonos, como en la figura 3. El tipo a utilizar depende del número de posibles salidas para cada lugar.



En las aventuras más sencillas, sólo se podrá salir de cada lugar en una de cuatro direcciones: Norte, Sur, Este y Oeste (así ocurre en la aventura del globo ocular). Si tu aventura es de este tipo, tienes que transferir a la rejilla cuadrada toda la información para que resulte equivalente al mapa. Más adelante desarrollaremos esto con detalle. Si has utilizado salidas que incluyen Nordeste, Noroeste, Sureste y Suroeste, tienes que emplear la retícula octogonal, aunque este tipo de sistema de rejillas resulta muy complicado.

La última variante de la rejilla se produce si has decidido también subir y bajar. En este caso la mejor solución es utilizar una rejilla separada para cada «nivel» de aventura.

La aventura del ojo precioso está basada sobre una retícula cuadrada, permitiendo sólo salidas por el Norte, Sur, Este y Oeste. A menos que haya una necesida real de otras direcciones, este tipo de aventuras resulta muy adecuado y hay una manera de obviar las subidas y bajadas. Si utilizas una descripción con frases como

HAY UNA ESCALERA QUE BAJA HACIA EL OESTE

puedes usar la respuesta normal del Oeste o la rutina adecuada.

LA RETICULA

Esta aventura requiere una retícula cuadrada de 6×4 como puedes comprobar contando el número máximo de lugares de tu mapa, de arriba a abajo y de derecha a izquierda. Antes de empezar a trasladar todos los detalles a la retícula, asegúrate de que has numerado todos los cuadrados. Empieza con el número 1 en la parte superior izquierda, y prosigue hasta llegar a la parte inferior derecha. Cuando hayas numerado todos los cuadrados y transferido todos los detalles, la retícula será como la de la figura 4.

EMPIEZA EL PROGRAMA

Ahora tienes una línea histórica y una retícula completa, puedes empezar con el programa.

El primer paso es teclear las descripciones de los lugares, basándote en tu retícula. Tienes que decidir cómo van a ser de largas. Intenta sugerir lo más posible el ambiente de la aventura sin derrochar memoria. Aparte de las descripciones de los lugares, hay que decirle al ordenador en qué direcciones se encuentranlas salidas.

Aquí tienes al fin las primeras secciones del programa. Los números de líneas tan altos te permitirán disponer de suficiente espacio en las anteriores secciones de programa a medida que vayas desarrollando el juego.

Teclea esta sección y almacenala (SAVE) en cinta:

Teclea para C-64 y Vic-20

```
5000 REM ** DESCRIPCION DE
     UBICACIONES
5010 REM ** UBICACION 4
5020 PRINT"ESTAS EN EL
     EXTERIOR DE UN GRAN
     EDIFICIO"
5030 N=0:E=0:S=1:O=0:RETURN
5040 REM ** UBICACION 7
5050 PRINT"ESTAS EN UN RIO
     TURBULENTO"
5060 N=0:E=1:S=0:O=0:RETURN
5070 REM ** UBICACION 8
5080 PRINT"ESTAS EN UN
     BOSQUE PETRIFICADO"
5090 N=0:E=0:S=1:O=1:RETURN
5100 REM ** UBICACION 10
5110 PRINT"ESTAS UNA SALA
     POLVORIENTA"
5120 N=1:E=1:S=1:O=0:RETURN
5130 REM ** UBICACION 11
5140 PRINT"ESTAS UN CUARTO
     OSCURO"
5150 IF OB(6)<>-1 OR LA<>1
     THEN N=0:E=0:S=0:O=0
5155 IF OB(6)<>-1 OR LA<>1
     THEN PRINT"NO SE VEN
     LAS SALIDAS CON LA
     OSCURIDAD"
5156 RETURN
5160 N=0:E=0:S=1:O=1:RETURN
5170 REM ** UBICACION 14
5180 PRINT"ESTAS EN UN CAMINO
     LLENO DE BARRO"
5190 N=1:E=1:S=0:O=0:RETURN
5200 REM ** UBICACION 15
5210 PRINT"ESTAS EN LA PUERTA
     DE LA CIUDAD OCULTA"
5220 N=0:E=1:S=0:O=1:RETURN
5230 REM ** UBICACION 16
5240 PRINT"ESTAS EN EL HALL
     DE ENTRADA"
5250 N=1:E=1:S=1:O=1:RETURN
5260 REM ** UBICACION 17
5270 PRINT"ESTAS EN EL PATIO"
5280 N=1:E=1:S=0:O=1:RETURN
5290 REM ** UBICACION 18
5300 PRINT"ESTAS EN EL
     JARDIN"
5310 N=0:E=0:S=1:O=1:RETURN
5320 REM ** UBICACION 22
5330 PRINT"ESTAS EN EL
     ARMARIO"
5340 N=1:E=0:S=0:O=0:RETURN
5350 REM ** UBICACION 24
5360 PRINT"ESTAS EN LA SALA
     DEL TRONO"
5370 N=1:E=0:S=0:O=0:RETURN
```

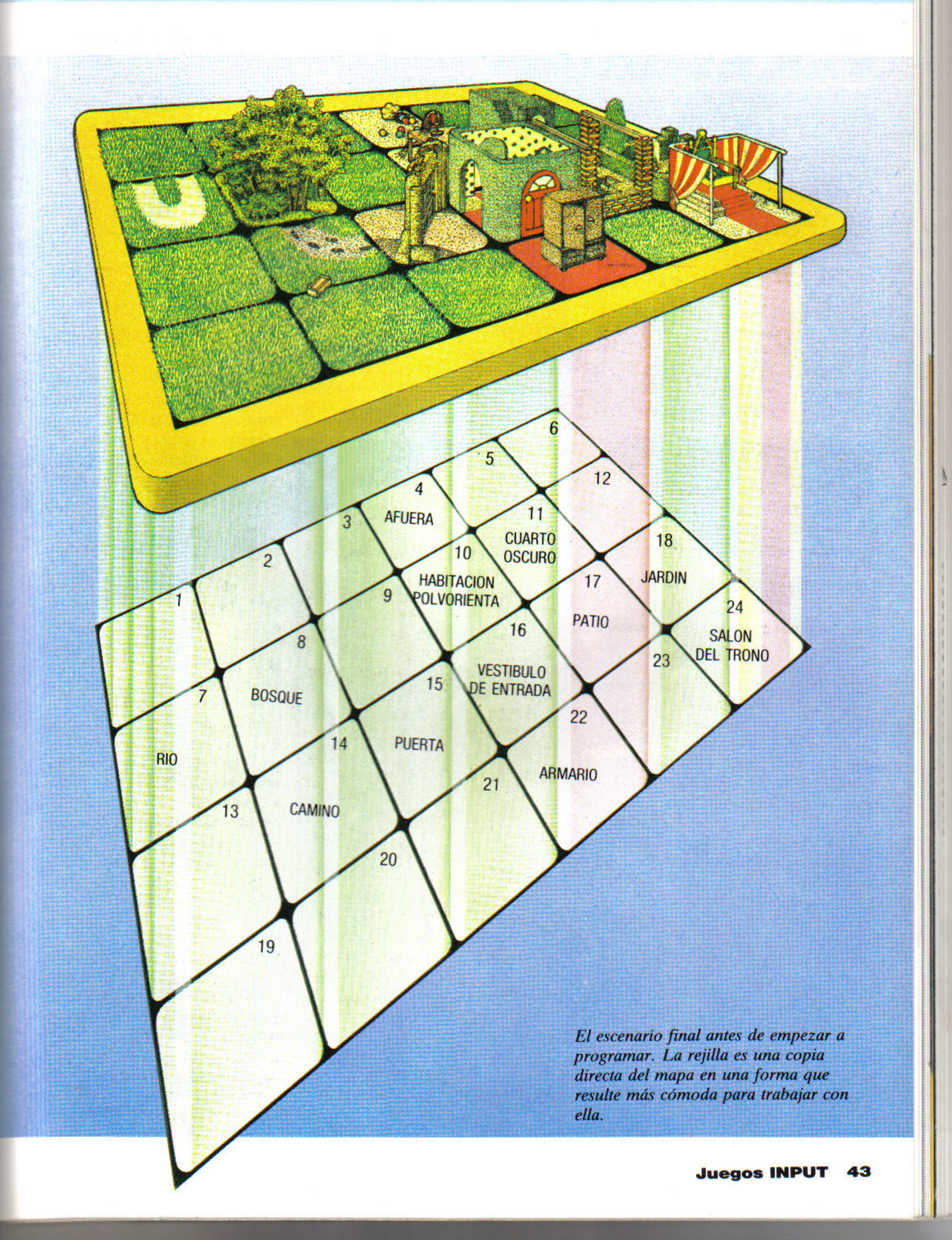
No te preocupes por el abundante uso de las sentencias REM y toda la memoria que consumen. En esta primera etapa del desarrollo del programa es muy importante saber qué es lo que hace cada parte del programa, o a qué número de lugar se refiere una descripción particular. Siempre puedes quitarlas más adelante.

Después de cada línea con la descripción de un lugar, hay otra línea que contiene información sobre sus posibles salidas. Las variables N, S, E y O significan Norte, Sur, Este y Oeste. Estas variables pueden tomar uno de dos valores posibles: 0 significa que no hay salida en esa dirección, mientras que 1 significa que hay salida.

Finalmente, después de cada sección de este programa hay un RETURN, debido a que cada descripción de lugar se llama mediante una sentencia GOSUB.

La sentencia IF ... THEN que hay en la sección del cuarto oscuro es para comprobar si el aventurero tiene la lámpara, pero la descripción de las variables se comentará más adelante cuando nos ocupemos de los objetos.

En el próximo capítulo vermos la manera en que se mueve el aventurero por los distintos lugares.



El escenario final antes de empezar a programar. La rejilla es una copia directa del mapa en una forma que resulte más cómoda para trabajar con ella.

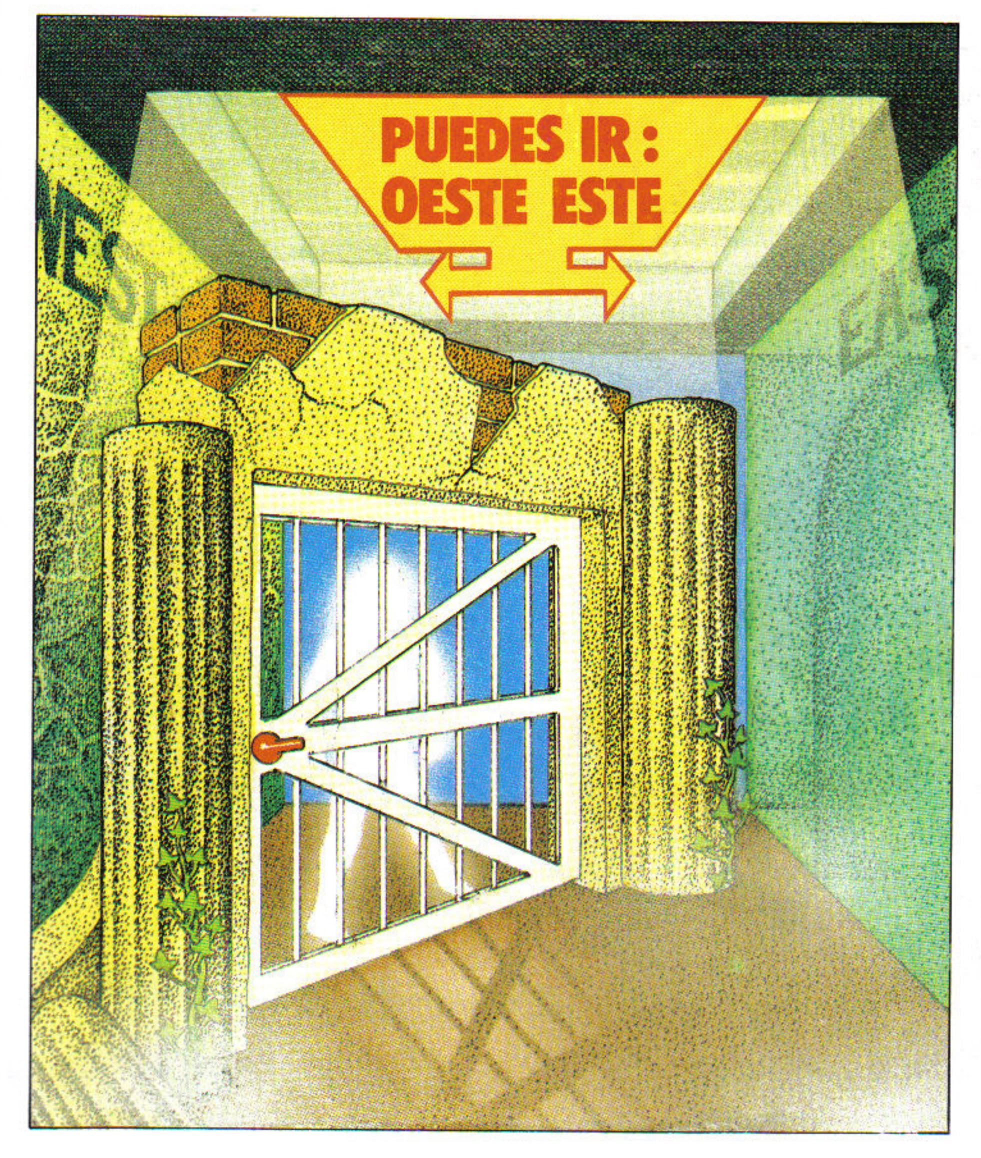
UNA AVENTURA MOVIDA

- PRESENTACION DE DIRECCIONES Y DESCRIPCIONES
- TRATAMIENTO DE LAS INSTRUCCIONES
- BLOQUEO DE LOS MOVIMIENTOS IMPOSIBLES

Una parte de la diversión de jugar juegos de aventuras es la posibilidad de explorar mundos nuevos y extraños sin salir de casa. Veamos la manera de iniciar estas exploraciones.

Ahora que ya has tecleado un programa que contiene todas las descripciones de los ambientes, es el momento de hacer que el aventurero pueda explorarlos, desplazándose de un lugar a otro. Tienes que prever todos los movimientos posibles en cada punto, y fijar unos límites, basándote en el grado de dificultad y en las pistas que se van recogiendo a medida que progresa la aventura.

En esta ocasión, para ampliar el desarrollo de tu programa de aventuras, presentaremos rutinas con la descripción correcta de los ambientes, junto con las posibles salidas a los mismos. Al jugador se le pedirá que introduzca respuestas, y verás la manera de escribir una sección de programa para que el jugador se vaya moviendo por el interior del fantástico mundo que hemos creado, en base a las elecciones hechas, a medida que progresa la aventura.



EL PUNTO DE PARTIDA

Lo primero que el ordenador tiene que saber es dónde está el aventurero en cada momento del juego. Para hacer esto el programa mira al valor que va tomando una variable llamada L (inicial de lugar). A esta variable se le asigna un valor que corresponde al ambiente en que se encuentra el aventurero después de realizar cada movimiento.

Para empezar pues la aventura, tienes que decirle al ordenador dónde quieres que se encuentre el aventurero en el momento de empezar.

Aquí tienes la primera parte de un programa que se encarga de esto. Carga (con LOAD) la sección que tecleaste la última vez, y añádele las siguientes líneas:

Teclea para C-64 y Vic-20

```
270 REM ** POSICION DE
    COMIENZO
280 L=15
290 GOTO 330
```

El número 15 corresponde al lugar en que se encuentra la puerta de la ciudad escondida. Si quieres que la aventura empiece en otro lugar, basta con que modifiques el valor de L. En seguida veremos la forma de modificar el valor de L durante el juego, según el lugar atravesado. Pero antes de que el aventurero pueda empezar a moverse, hay que decirle al ordenador hacia dónde tiene que ir.

RESPUESTAS

Para que el ordenador pueda entender adecuadamente tus respuestas, y actuar en consecuencia, tienes que proporcionarle una lista de todas las palabras que puede reconocer.

En esta etapa del desarrollo, basta con que reconozca las cuatro direcciones correspondientes a los cuatro puntos cardinales: NORTE, SUR, ESTE y OESTE. Podemos meterlas en una matriz R$, que contiene los datos de cada dirección de respuesta.

Teclea para C-64 y Vic-20

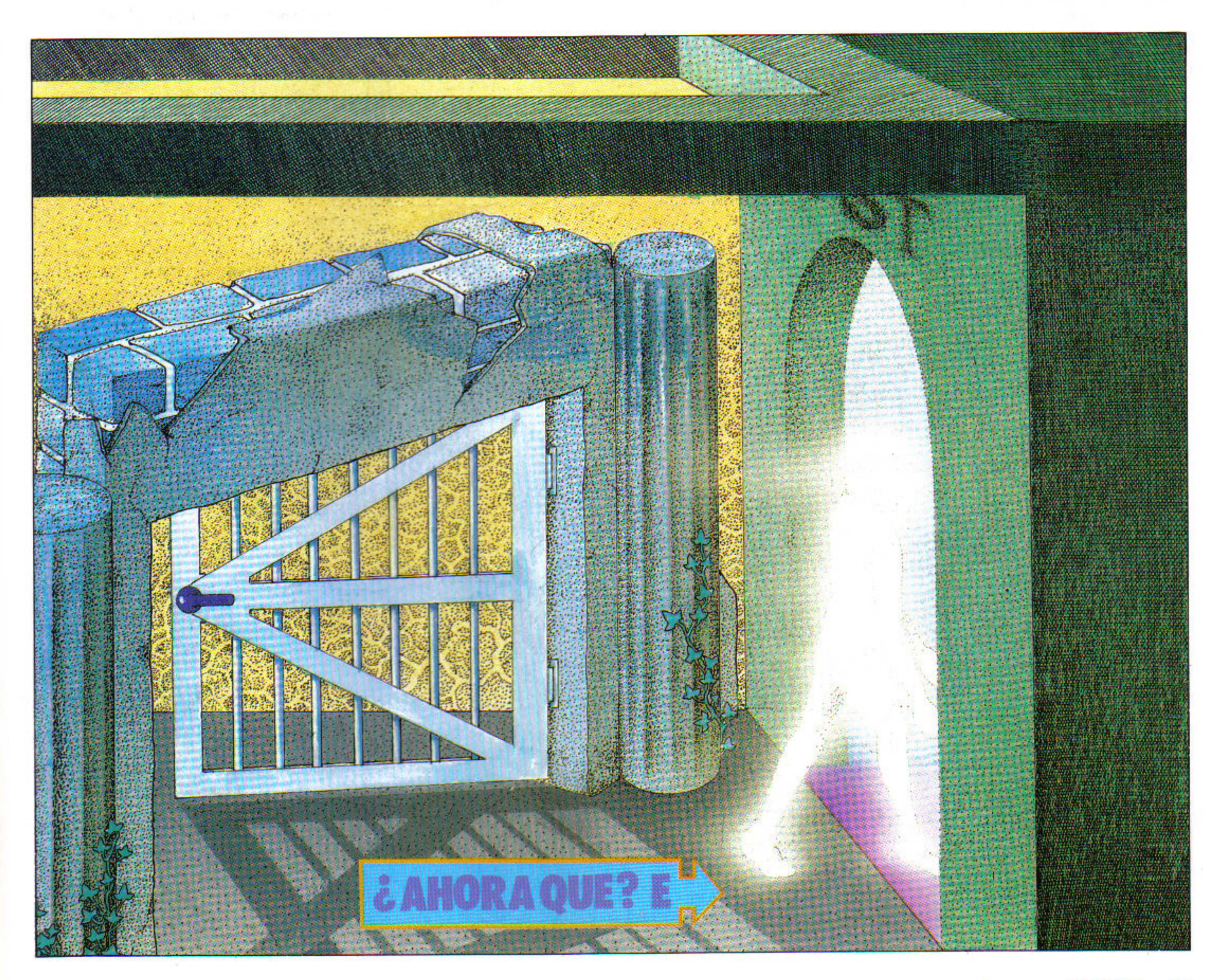
```
110 REM ** AJUSTE DE CADENAS
    PARA RESPUESTAS
120 DIM R$(19),R(19)
130 FOR K=1 TO 4:READ R$(K),
    R(K):NEXT
150 DATA NORTE,1,SUR,1,ESTE,
    1,OESTE,1
```

En la línea 120 se dimensionan las matrices, de modo que contengan todas las respuestas necesarias para el juego. Dado que por el momento sólo necesitas utilizar las direcciones, sólo se emplearán los cuatro primeros elementos de las matrices R$ y R. El bucle FOR ... NEXT de la línea 130, va desde uno hasta cuatro, leyendo tanto en R$ como en R. Las direcciones y sus números están en las sentencias DATA de la línea 150.

Pero evidentemente esta información no le sirve para nada al jugador, a menos que el ordenador le diga previamente dónde se encuentra.

LA BUSQUEDA DE UN LUGAR

Para que el aventurero pueda saber adonde ha ido a parar después de cada uno de sus movimientos, hay que suministrarle una descripción de los lugares. Estas descripciones de lugares ya las tienes tecleadas; sólo te hace falta una rutina para extraer la descripción que corresponda al valor de la variable L, el número de lugar. Aquí es donde te resultarán muy útiles las líneas REM que introdujiste anteriormente.






```
300 REM **ENCONTRAR
    UBICACION**
310 PRINT"[SHIFT+CLR/HOME]"
330 IF L<11 THEN ON L GOSUB
    0,0,0,5020,0,0,0,5050,
    5080,0,5110:GOTO 400
340 IF L<21 THEN ON L-10
    GOSUB 5140,0,0,5180,5210
    ,5240,5270,5300,0,0:
    GOTO 400
350 IF L<26 THEN ON L-20
    GOSUB 0,5330,0,5360
```

Antes de escribir este tipo de rutinas, tienes que asegurarte del número correspondiente a cada descripción de lugar. A partir del lugar número 1, tienes que ir haciendo una lista de los números de líneas de cada descripción. Si para alguno de los ambientes no hay ninguna descripción especial, asígnale un 0. En nuestra aventura, por ejemplo, no hay descripciones para los lugares 1, 2 y 3, pero sí la hay para el lugar 4.

Ahora que tienes la lista de los números de líneas, puedes empezar a escribir la rutina. Es una sencilla secuencia de operaciones de control del valor de L por medio de una sentencia ON ... GOSUB.

La lista de números de líneas va desde la línea 330 a la 350, comenzando por el lugar 1 al principio de la línea 330 y terminando con el lugar 24 al final de la línea 350.

PRESENTACION DE LAS DIRECCIONES

Aparte de conocer las descripciones de los lugares, el jugador de la aventura querrá saber qué salidas tiene. Como en cada lugar no le será posible moverse en todas direcciones, el programa debe tener alguna forma de controlar las informaciones asociadas con la dirección, es decir las variables

N, S, E y O. Las siguientes rutinas dirán al aventurero qué direcciones son posibles:

```
390 REM ** MUESTRA DIRECCION
400 IF L<>11 OR (LA=1 AND
    OB(6)=-1)THEN PRINT:
    PRINT"[CTRL+4]PUEDES IR
    [CTRL+8]";:GOTO 410
405 GOTO 460
410 IF N>0 THEN PRINT TAB
    (11);"NORTE"
420 IF E>0 THEN PRINT TAB
    (11);"ESTE"
430 IF S>0 THEN PRINT TAB
    (11);"SUR"
440 IF O>0 THEN PRINT TAB
    (11);"OESTE"
```

La rutina se limita a comprobar el valor de las variables N, S, E y O, basándose en el mapa de direcciones. Si el valor de las variables es mayor que cero, la dirección es posible y se presenta la correspondiente salida.

Esta rutina puede ser incorporada tal como está en cualquier aventura basada en una retícula de cuadrados.

INSTRUCCIONES

Ahora que el aventurero ya conoce las direcciones que puede tomar, hay que hacerle algunas sugerencias. La siguiente sección de programa le preguntará al jugador ¿QUE VAS A HACER AHORA?

Teclea para C-64 y Vic-20

```
450 REM **INSTRUCCIONES
460 PRINT:INPUT"[CTRL+2]
    AHORA QUE[CTRL+8]";I$
470 GOSUB 3010
```

Se trata de una rutina de entrada muy sencilla. A la respuesta suminis-



trada la llamamos I$. El ordenador debe comprobar que la respuesta es correcta y actuar en consecuencia. La línea 470 envía el control a la subrutina de la línea 3010, que es la encargada de procesar la respuesta dada por el jugador.

Teclea para C-64 y Vic-20

```
3000 REM ** COMPRUEBA
     INSTRUCCION
3010 N$="":FOR Z=1 TO LEN
     (I$):IF MID$(I$,Z,1)=
     " "THEN I=Z:GOTO 3020
3015 NEXT:I=0
3020 IF I=0 THEN V$=I$:GOTO
     3050
3030 V$=LEFT$(I$,I-1)
3040 N$=MID$(I$,I+1)
3050 I=0
3060 FOR K=1 TO 19
3070 IF V$=LEFT$(R$(K),LEN
     (V$)) THEN I=R(K):I$=
     LEFT$(V$,1)
3080 NEXT
3090 RETURN
```

Esta rutina comprueba si I$ contiene dos palabras. De ser así, llama V$ a la primera y N$ a la segunda. Hemos puesto V$ por verbo, como COGER, MATAR, o LLEVAR, y también sirve para las direcciones como NORTE, SUR, ESTE Y OESTE. N$ significa nombre, y vale para designar los objetos que aparecen en la aventura. El programa utiliza la sentencia MID$ (líneas 3010 a 3015) para examinar si hay un espacio entre V$ y N$.

Si se encuentra un espacio, la línea 3020 designa a las dos partes de I$ con las etiquetas N$ y V$. Si no se encuentra espacio, la propia línea 3020 llama V$ a toda I$.

El resto de la subrutina (líneas 3060 a 3080) compara las respuestas obtenidas con las que contiene la matriz R$, que es el que contiene las direcciones de las respuestas. Más adelante veremos cómo puede ampliarse para contener una serie adicional de verbos. Si en la línea 3070 se detecta una coincidencia, se pone en I el correspondiente valor de R. En fases posteriores del programa la máquina puede saber si ha habido una coincidencia examinando si I es mayor que cero. La última parte de la línea 3070, toma la primera letra de V$ y la llama I$, que servirá después para el movimiento del aventurero.

Esta subrutina puede utilizarse casi sin limitaciones en cualquier juego de aventuras. Probablemente la única modificación que le hagas, estará relacionada con la longitud del bucle FOR ... NEXT de la línea 3060.

MOVIMIENTO

Para que el aventurero pueda explorar todos los lugares, basta que añadas al programa una rutina que transforme la variable de lugar L, según el contenido de I$. Aquí la tienes:

Teclea para C-64 y Vic-20

```
1000 REM ** RUTINA DE
     MOVIMIENTO
1010 IF I$="N" AND N>0 THEN
     L=L-6:GOTO 310
1020 IF I$="E" AND E>0 THEN
     L=L+1:GOTO 310
1030 IF I$="S" AND S>0 THEN
     L=L+6:GOTO  310
1040 IF I$="O" AND O>0 THEN
     L=L-1:GOTO 310
1050 REM ** SI NO HAY
     UBICACION POSIBLE EN TAL
     DIRECCION
1060 PRINT:PRINT"LO SIENTO,
     NO PUEDES IR ES ESTA
     DIRECCION!!":GOTO 330
```

Recuerda que la aventura estaba basada en una retícula de seis lugares. El movimiento implica un cambio del valor de L, que depende del tamaño de la retícula. Moverse hacia el Norte o hacia el Sur equivale a sumar o restar 6 al valor de L para desplazarse una línea hacia arriba o hacia abajo en la retícula. Análogamente, el movimiento hacia el Este o hacia el Oeste es equivalente a sumarle o restarle 1 al valor de L.

Las líneas 1010 a 1040 comprueban las direcciones en I$ y ajustan L en consecuencia. Sólo es posible el movimiento si hay una salida que coincide con I$. Las salidas fueron definidas en las líneas inmediatamente siguientes a las descripciones de los lugares.

Si no existe salida en la dirección en que el aventurero quiere salir, la línea 60 presenta el mensaje ¡LO SIENTO! ¡NO PUEDES SALIR POR AHI!

Para utilizar esta rutina con una aventura diferente, el único cambio que tendrás que hacer tendrá que ver con el tamaño de la retícula. En tal caso, habrá que modificar las líneas 1010 a 1030, con arreglo a las líneas de la retícula.

Almacena ahora el programa (con SAVE) dejándolo listo para ser utilizado en del próximo capítulo.

P y R

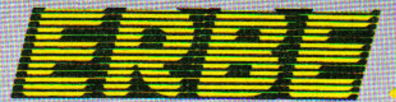
¿Entenderá el programa respuestas tales como NORTE o IR NORTE, además de letras solas, tal como N?

R: La rutina de comprobación de instrucciones, situada en las líneas 3000 a 3010, está escrita de forma que recibe palabras de una sola letra, a las que maneja separadamente de las otras instrucciones más largas.

Las instrucciones de una sola letra se llaman I$ y más tarde en el programa se comprueba si I$ contiene las letras N, S, E o W, permitiendo al jugador que responda con la forma abreviada de la instrucción, y haciendo que el juego resulte más rápido de jugar.

Sin embargo, no hay nada que te impida hacer adiciones que permitan al aventurero utilizar respuestas del tipo NORTE o IR NORTE. La próxima vez veremos cómo maneja el programa los verbos y los nombres.

Lo que hay que hacer es añadir en la línea las palabras completas de las direcciones o añadir IR a la lista de verbos y escribir una rutina de procesamiento de los nuevos verbos.

SI BUSCAS LO MEJOR  LO TIENE



ROCK'N LUCHA

MELBOURNE HOUSE

DISTRIBUIDOR EXCLUSIVO PARA ESPAÑA ERBE SOFTWARE, SANTA ENGRACIA, 17. Tel: 447 34 10. DELEGACION BARCELONA, Avd. MISTRAL, 10. Tel. (93) 432 07 31

EL DISEÑO DE TARJETAS HECHO FACIL

Diseñar las pistas de cobre que darán un significado coherente al conjunto de componentes que se aglutina sobre una tarjeta, es una de las tareas más ingrantas para los aficionados y profesionales de la electrónica.

Los técnicos de la especialidad han buscado siempre nuevos métodos que aligeren la pesadez de este obligado trabajo, heredero directo de los incómodos y poco fiables circuitos «cableados».

El advenimiento del CAD (diseño asistido por ordenador) trajo consigo los primeros logros en este terreno, produciendo resultados satisfactorios y, a veces, sorprendentes. Como todo lo bueno, también existían inconvenientes, siendo el principal su elevado precio y la necesidad de sistemas potentes y caros, convirtiéndolo en un recurso prohibitivo para la mayoría de los potenciales usuarios.

Lo anterior fue cierto durante algún tiempo, pero en la actualidad han aparecido algunos programas destinados a microordenadores de bajo precio, tal como es el caso del **Commodore 64.**

A continuación pasamos a describir un programa desarrollado por la firma alemana **Data Becker** y traducido al castellano por **Ferre-Moret.** Con él se mecaniza el diseño de circuitos impresos por aproximadamente treinta mil pesetas.

El programa en cuestión se llama **Platine 64** y puede clasificársele como herramienta de diseño.

El primer paso obligado es disponer de un esquema teórico en el que están representados los componentes necesarios y las conexiones que forman un circuito con todos ellos.

El programa se carga desde la unidad de *diskettes* a la memoria central del **64,** apareciendo poco después en la pantalla el consabido menú de opciones.

El menú principal permite elegir distintas opciones de trabajo. En la pantalla aparece una serie de puntos formando una retícula sobre la cual habrá que ir volcando el diseño. Los puntos aparecen colocados en hileras regulares, siendo la distancia entre unos y otros de 1,27 mm, una distancia estandarizada para el desarrollo de tarjetas.

La superficie total de la placa es mayor de lo que puede verse en la pantalla, y es tan solo una ventana por la que asoma parte de la matriz total, formada por 127 puntos en sentido horizontal y 78 en vertical.

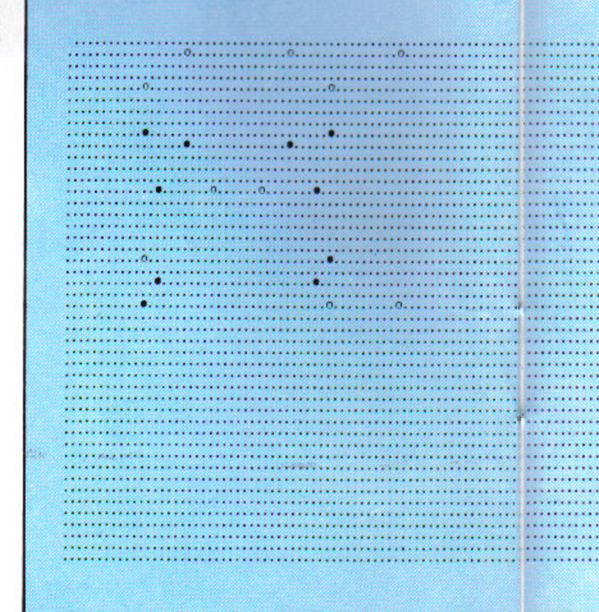
El trabajo inicial habrá de realizarse en base a la opción manual (tecla M), pudiendo acceder a las dos caras de la tarjeta, definidas como soldadura y componentes. La Tecla F1 permite cambiar de cara instantáneamente.

La fijación de terminales (circulitos) en la placa se efectúa con la tecla F5; previamente se ha colocado el cursor sobre el punto que nos interesa. El desplazamiento por la pantalla puede realizarse utilizando las teclas de cursor o, preferentemente, el *joystick.* En la base de la pantalla se muestran las coordenadas del punto sobre el que está situado el cursor en cada momento.

En este modo de trabajo es posible, mediante la pulsación de solo una tecla, borrar toda una cara (F4), Borrar trozos de una línea (L), Borrar un conductor (F6), trazar (Z), etc.

Un interesante recurso que ofrece el **Platine 64** es el acceso directo a los menús de ayuda sin tener que abandonar el tablero de dibujo, puesto que aparecen en la misma pantalla de trabajo.

La vista de los terminales que vamos situando sobre la matriz de puntos puede parecer anodina en un momento dado y llevarnos a confusiones, el problema lo remedia la tecla F3,

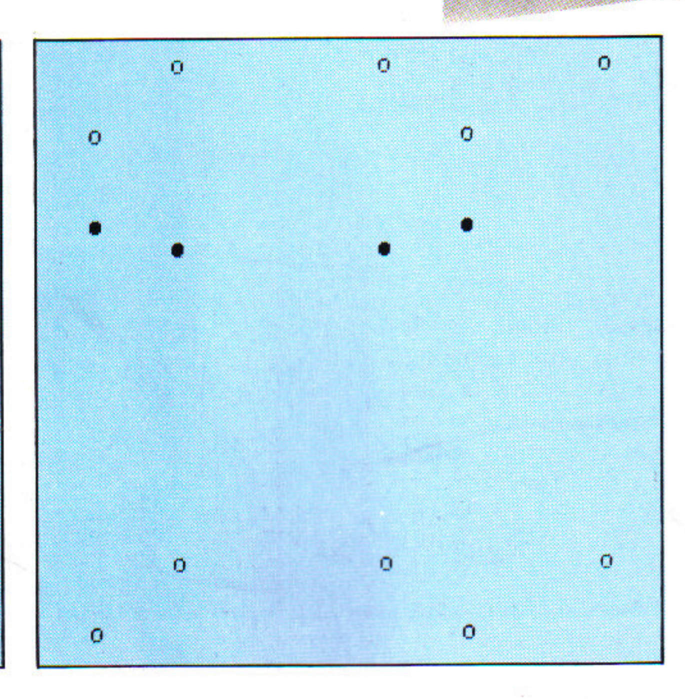


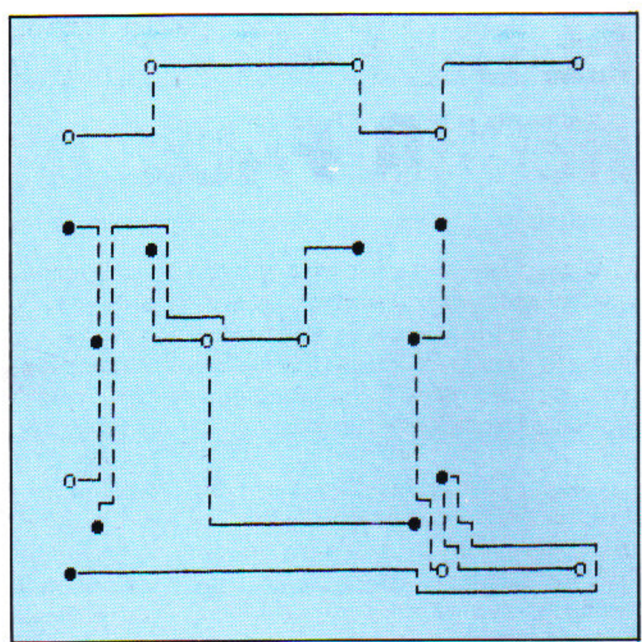
que visualiza en la «línea de estado» la leyenda del componente o terminal al que fue destinado, algo así como si hubiéramos escrito una referencia con un bolígrafo cuando este trabajo se hace simplemente con papel y lápiz.

Las teclas 1, 2 y 3 sirven para volcar ambas caras directamente sobre la impresora, pudiendo elegirse que aparezca o no simultáneamente la matriz de puntos que vemos siempre en la pantalla.

Introducir cada uno de los terminales de cada componente, cada vez que lo precisemos, es una tarea tediosa y que puede llevar a errores. En solución a tal inconveniente, el programa recurre a la unidad de *diskettes.* La creación de ficheros de componentes específicos es el remedio. En esta opción, con la tecla F4, se pueden describir todos los componentes que sean precisos. Sobre la matriz de puntos se describen los terminales y su disposición, teniendo en cuenta que habrán de situarse sobre los puntos separados 1,27 mm consecutivamente.

Antes de utilizar cualquiera de los componentes almacenados en *diskette* es obligado abrir el fichero correspondiente y a partir de entonces es posible cargarlos en memoria uno a uno o buscar uno determinado. El menú para la biblioteca de componentes (F8) resulta de gran ayuda mientras





se familiariza el usuario con esta opción.

Paralelamente al trabajo de diseño, puede obtenerse una lista de los componentes y terminales que se han dispuesto sobre las superficies de la tarjeta.

Un componente puede ser colocado en cualquier parte de la tarjeta, pero si alguno de sus terminales se superpone a otro ya definido, el hecho aparece reflejado mediante un mensaje de error. Los componentes pueden igualmente ser girados 90 grados con sólo presionar una tecla.

Una vez que se han decidido qué componentes irán depositados en la tarjeta y sus posiciones en ella, hay que elaborar una lista de uniones entre terminales, que se describirán detalladamente a medida que el programa lo exige. Después se puede obtener una lista de todas las uniones definidas, para verificar que no se han introducido errores. De hecho es el propio programa quien indica si hemos dejado algún terminal por conectar.

Tal vez la posibilidad más interesante y potente que aporta el **Platine 64** es la opción llamada *Autoruter,* que se encarga precisamente de dibujar automáticamente las rutas que seguirán las pistas de cobre y además tiene criterios propios. Por ejemplo, si dos terminales no pueden ser unidos directamente, él por su cuenta trata de unir entre sí terminales que a su vez están unidos a éste.

A la hora de dibujar las pistas sobre el papel existen otras alternativas a la impresora por matriz de puntos. Se puede utilizar la impresora/*plotter* **VIC 1520** de **Commodore.** También se puede recurrir a los *interfaces* estándar RS-232 o Centronics. Las coordenadas de los taladros que deberán efectuarse una vez acabado el proceso fotoquímico a que se somete la tarjeta de circuito impreso, pueden obtenerse detalladamente en una lista que

```
COMPONENTES DEL FICHERODEMO :

  NOMBRE:          TERMINALES:

   TR          ...........003
   R           ...........002
   C           ...........002
   GL          ...........002
```

```
CONTENIDO DE LA LISTA DE COMPONENTES:

  1        .TR        003 P
  2        .TR        003 P
  3        .C         002 P
  4        .C         002 P
  5        .GL        002 P
  6        .GL        002 P
  7        .R         002 P
  8        .R         002 P
```

```
                    LISTA DE UNIONES:

1] 5          /2      .. R      1] ????????/          R
2] ????????/                    2] 4         /-    ..
1] 4          /-      .. R      1] ********/+      .. R
2] 6          /2      ..        2] 6         /1    ..
1] ????????/             R      1] ********/-      .. R
2] 7          /2      ..        2] ????????/
1] 8          /2      .. R      1] 7         /1    .. R
2] 4          /+      ..        2] 8         /1    ..
1] ????????/             R      1] ????????/          R
2] ????????/                    2] ????????/
1] 3          /+      .. R      1] 4         /+    .. R
2] 7          /1      ..        2] ????????/
1] 6          /1      .. R      1] ????????/          R
2] 8          /1      ..        2] ????????/
```

produce el **Platine 64,** de tal manera que con estos datos podría alimentarse una máquina herramienta automática que realice esta función.

El programa viene complementado con un manual que explica detenidamente todas las posibilidades en la primera parte. Después existe una sección que conduce al usuario a través de un ejemplo de utilización, que desvela algunos aspectos que no quedan claros en la primera parte.

La utilización de una herramienta de diseño de estas características permite solventar muchas de las tediosas complicaciones que surgen durante la creación de una tarjeta de circuito impreso por simple que sea éste.

PROGRAMACION EN CODIGO MAQUINA

- BUSCANDO UN LUGAR PARA EL CODIGO MAQUINA
- MONITORES DE CODIGO MAQUINA
- USO DEL MONITOR
- EJECUCION DEL PROGRAMA

Cuando enciendes tu ordenador, éste se va automáticamente al BASIC. Por eso tienes que usar un programa en BASIC para POKEar de tu programa en código máquina y un comando BASIC para ejecutarlo.

Al entrar en el maravilloso mundo de la programación en lenguaje máquina, te encontrarás con bastantes paradojas extrañas. Aunque el código máquina es el lenguaje realmente utilizado por el propio microprocesador, no puedes teclearlo directamente en tu ordenador doméstico. Tienes que introducirlo a través de otro programa, lo que en tu micro es equivalente a utilizar el BASIC. Tampoco puedes ejecutar un programa en código má-



quina de una forma inmediata. Tienes que llamarlo desde un programa o utilizar una simple instrucción del BASIC.

Para introducir una rutina en código máquina debes hacerlo mediante POKEs, byte a byte, en la memoria del ordenador. Paradójicamente en los **Commodore** no se puede POKEar un número hexadecimal, sino sólo números decimales. Por eso, los números hexadecimales de tus rutinas en código máquina tienes que pasarlos a decimales antes de introducirlos.

DONDE PONERLO

Antes de introducir un programa en lenguaje máquina, tienes que decidir dónde lo vas a ubicar. Esta es una cuestión muy importante. Evidentemente no lo puedes poner en una zona que utilice el propio ordenador, ya que sobre tu programa se escribirían nuevos datos procedentes de la ROM, y el ordenador podría funcionar indebidamente, por haber sido borrados datos vitales.

También tienes que ser precavido al utilizar el área reservada para el BASIC. Como tu rutina en código máquina puede ser llamada desde un programa en BASIC, podrían escribirse unas cosas encima de otras, haciendo que todo deje de funcionar.

Con programas pequeños puedes utilizar zonas como el *buffer* destinado a la cinta, suponiendo que no pienses utilizar la cinta para otra cosa.

En el **Commodore 64** la zona de memoria desde $C000 a $CFFF (en hexadecimal), o lo que es lo mismo desde 49152 a 53247 (en decimal), está reservada para código máquina, y este área cubrirá normalmente las necesidades para tus rutinas en código máquina. En el **Vic 20** no hay área reservada.

Se puede almacenar pequeños programas —de menos de 191 bytes— en el *buffer* de cinta, tanto en el **64** como en el **Vic**, suponiendo, naturalmente que no quieras usar *cassette* para alguna otra cosa. El *buffer* de cinta se extiende desde $033C a $03FB, ó 828 a 1019 en decimal.

Pero si te hace falta más espacio, puedes limitar el área de BASIC y dejar un área protegida sobre ella, igual que hace el comando CLEAR en otras máquinas. En el **Commodore 64** esto se hace con POKE en 52 y 56, con el valor 32, por ejemplo. Así se deja limpio hasta 8192, quedando 32768 bytes que no pueden sufrir la sobreescritura del BASIC y pueden utilizarse para las rutinas en código máquina, quedando unos 6144 bytes libres para los programas en BASIC.

Después de hacer esto, teclea CLR, es decir, las letras C, L y R separadamente. No pulses la tecla CLR, ya que si lo haces, cualquier programa que tengas en la zona de BASIC es probable que deje de funcionar.

Las posiciones de memoria 52 y 56 son los bytes altos de los dos puntos que especifican la parte superior del área de BASIC. Los bytes bajos, 51 y 55, son normalmente 0. Así 52 y 56 especifican en qué página termina el BASIC. Si haces en ellos un POKE de 32, se desplazará el puntero a 32 × 256 = 8192.

Puedes eliminar completamente el área del BASIC, haciendo en 52 y 56 un POKE de 8, lo que sitúa la parte superior del BASIC en 2048, que es precisamente el fondo del área de BASIC. Incluso puedes hacer en 52 y 56 un POKE con 0. Pero en estos casos el **Commodore** no te servirá de gran cosa, ya que no podrás hacer nada en BASIC y no podrás llamar a las rutinas en código máquina.

También puedes desplazar hacia arriba el final del área de BASIC y proteger tus rutinas colocándolas a continuación. Esto se hace con POKE en las direcciones de memoria 43 y 44. Se trata de los bytes alto y bajo de los punteros que especifican el fondo de la zona del BASIC. También aquí puedes desplazar el fondo de la zona de BASIC a 8192, con un POKE de 1 en la posición 43 y de 32 en la posición 44. También tienes que hacer POKE de 0 en el nuevo comienzo del BASIC, posición 8192. Esto dejará abundante espacio para rutinas en código máquina debajo del BASIC.

Después de hacer todo esto, tienes que teclear NEW, ya que en caso contrario cualquier programa que tengas en la zona de BASIC se verá confundido y se romperá.

La magnitud del desplazamiento del RAMTOP en el **Vic** depende de lo que se haya ampliado la memoria. Teclea PEEK 44 para ver donde está el final de la zona del BASIC y a continuación haz un POKE en 52 y 56 con ese número aumentado en 1 ó 2, para dejar sitio para un poco de BASIC.

MONITORES DE CODIGO MAQUINA

Los programas que siguen te permitirán introducir programas en lenguaje máquina, almacenarlos en cinta y examinarlos en memoria. Antes de meter el programa tienes que hacer una «limpieza». Una vez que lo hayas hecho, hayas introducido el programa y pulsado RUN, se te pedirá que des una dirección de comienzo.

El programa ha de empezar en $C000, a menos que necesite más de 4095 bytes. En este caso tendrías que ponerlo en la zona de BASIC mediante un POKE de 32 en las direcciones 52 y 56, y situar el comienzo de tu rutina en código máquina en 8192 (decimal). En el **Vic 20**, el punto de comienzo del programa en código máquina depende de lo que se haya ampliado la memoria (ver más arriba). Teclea a continuación el monitor de código correspondiente a tu máquina.

Teclea para Commodore-64

```
5 POKE 53280,0:POKE 53281,0:
  HH$="0123456789ABCDEF"
10 PRINT"[SHIFT+CLR/HOME]
   [7*CRSR abajo][CTRL+4]"
   TAB(8)"1:- EDITA CODIGO
   MAQUINA"
20 PRINT TAB(8)"[2*CRSR
   abajo]2:- EXAMINA MEMORIA"
30 PRINT TAB(8)"[2*CRSR
   abajo]3:- GUARDA BYTES EN
   CINTA"
40 GET A$:IF A$<"1" OR A$>
   "3" THEN 40
50 ON VAL(A$) GOSUB 100,200,
   300
60 GOTO 10
```


2017 91 FB
2016 88
2014 B1 FB
2012 A0 01
2010 85 FD
200E B1 FB
200C A0 00
200A 85 FE
2008 A8 00
RAMTOP
1FF8 55 59
1FF7 2A
1FF6 4F
1FF5 4B
1FF2 2E 4D 4B
1FEF 2C 2F 2C
FE4 4B
FE1 4C 4C
EØ 4B

```
100 INPUT "[SHIFT+CLR/HOME]
    DIRECCION DE COMIENZO ";
    SA
110 INPUT"[CLR/HOME][2*CRSR
    abajo][6*ESPACIO][6*CRSR
    izq.]";D$:IF D$=""
    THEN 110
120 IF D$="#" THEN RETURN
125 W=0:FOR Z=1 TO 16:IF
    LEFT$(D$,1)=MID$(HH$,Z,1)
    THEN W=W+1
126 IF RIGHT$(D$,1)=MID$(HH$
    ,Z,1) THEN W=W+1
127 NEXT:IF W<2 THEN 110
130 A=ASC(D$)-48:B=ASC(
    RIGHT$(D$,1))-48
140 C=B+7*(B>9)-(LEN(D$)=2)*
    (16*(A++7*(A>9)))
150 POKE SA,C:PRINT"[CLR/
    HOME][5*CRSR abajo][2*
    ESPACIO][CRSR dcha.]"SA,
    D$
160 SA=SA+1:GOTO 110
200 INPUT "[SHIFT+CLR/HOME]
    DIRECCION INICIAL ";SA
210 INPUT"IMPRESORA (S/N) ";
    P$:PRINT"[SHIFT+CLR/
    HOME]"
220 IF P$="S" THEN OPEN 4,4:
    CMD 4
230 PRINT SA;:FOR M=0 TO 7
240 A=(PEEK(SA)/16):PRINT
    MID$(HH$,A+1,1);MID$(HH$,
    PEEK(SA)-INT(A)*16+1,1)
    " ";
250 SA=SA+1:NEXT
260 IF P$="S" THEN PRINT#4,
    "";:CLOSE 4
270 GET A$:IF A$="" THEN 270
280 IF A$=CHR$(13) THEN
    RETURN
290 PRINT:GOTO 220
300 CLR:INPUT "[SHIFT+CLR/
    HOME]ESCRIBE DIRECCION
    INICIAL";A:A=A-3:AA=INT
    (A/256):A2=A-AA*256
310 INPUT"ESCRIBE DIRECCION
    FINAL  ";B:B=B+1:BB=INT
    (B/256):B2=B-BB*256
315 INPUT "ESCRIBE NOMBRE
    FICHERO ";N$
320 PRINT "[SHIFT+CLR/HOME]
    P[SHIFT+O]44,"AA":P
    [SHIFT+O]43,"A2
330 PRINT "[2*CRSR abajo]P
    [SHIFT+O]46,"BB":P[SHIFT
    +O]45,"B2:PRINT "[2*CRSR
    abajo]SAVE"CHR$(34)N$
    CHR$(34)
340 PRINT "[5*CRSR abajo]P
    [SHIFT+O]44,"PEEK(44)":P
    [SHIFT+O]43,"PEEK(43)
350 PRINT "[2*CRSR abajo]P
    [SHIFT+O]46,"PEEK(46)":P
    [SHIFT+O]45,"PEEK(45):
    PRINT "[2*CRSR abajo]RUN"
```

Teclea para Vic-20

```
5 POKE 36879,8:HH$="01234567
  89ABCDEF"
10 PRINT"[SHIFT+CLR/HOME]
   [7*CRSR abajo][CTRL+4]1:-
   EDITA CODIGO MAQUINA"
20 PRINT "[2*CRSR abajo]2:-
   EXAMINA MEMORIA"
30 PRINT "[2*CRSR abajo]3:-
   GUARDA BYTES EN CINTA"
40 GET A$:IF A$<"1" OR A$>
   "3" THEN 40
50 ON VAL(A$) GOSUB 100,200,
   300
60 GOTO 10
100 INPUT "[SHIFT+CLR/HOME]
    DIRECCION DE COMIENZO ";
    SA
110 INPUT"[CLR/HOME][2*CRSR
    abajo][6*ESPACIO][6*CRSR
    izq.]";D$:IF D$=""
    THEN 110
120 IF D$="#" THEN RETURN
125 W=0:FOR Z=1 TO 16:IF
    LEFT$(D$,1)=MID$(HH$,Z,1)
    THEN W=W+1
126 IF RIGHT$(D$,1)=MID$(HH$
    ,Z,1) THEN W=W+1
127 NEXT:IF W<2 THEN 110
130 A=ASC(D$)-48:B=ASC(
    RIGHT$(D$,1))-48
140 C=B+7*(B>9)-(LEN(D$)=2)*
    (16*(A++7*(A>9)))
150 POKE SA,C:PRINT"[CLR/
    HOME][5*CRSR abajo][2*
    ESPACIO][CRSR dcha.]"SA,
    D$
160 SA=SA+1:GOTO 110
200 INPUT "[SHIFT+CLR/HOME]
    DIRECCION INICIAL ";SA
210 INPUT"IMPRESORA (S/N) ";
    P$:PRINT"[SHIFT+CLR/
    HOME]"
220 IF P$="S" THEN OPEN 4,4:
    CMD 4
230 PRINT SA;:FOR M=0 TO 4
240 A=(PEEK(SA)/16):PRINT
    MID$(HH$,A+1,1);MID$(HH$,
    PEEK(SA)-INT(A)*16+1,1)
    " ";
250 SA=SA+1:NEXT
260 IF P$="S" THEN PRINT#4,
    "";:CLOSE 4
270 GET A$:IF A$="" THEN 270
280 IF A$=CHR$(13) THEN
    RETURN
290 PRINT:GOTO 220
300 CLR:INPUT "[SHIFT+CLR/
    HOME]ESCRIBE DIRECCION
    INICIAL";A:A=A-3:AA=INT
    (A/256):A2=A-AA*256
310 INPUT"ESCRIBE DIRECCION
    FINAL  ";B:B=B+1:BB=INT
    (B/256):B2=B-BB*256
315 INPUT "ESCRIBE NOMBRE
    FICHERO ";N$
320 PRINT "[SHIFT+CLR/HOME]
    P[SHIFT+O]44,"AA":P
    [SHIFT+O]43,"A2
330 PRINT "[2*CRSR abajo]P
    [SHIFT+O]46,"BB":P[SHIFT
    +O]45,"B2:PRINT "[2*CRSR
    abajo]SAVE"CHR$(34)N$
    CHR$(34)
340 PRINT "[5*CRSR abajo]P
    [SHIFT+O]44,"PEEK(44)":P
    [SHIFT+O]43,"PEEK(43)
350 PRINT "[2*CRSR abajo]P
    [SHIFT+O]46,"PEEK(46)":P
    [SHIFT+O]45,"PEEK(45):
    PRINT "[2*CRSR abajo]RUN"
```

FUNCIONAMIENTO DEL MONITOR

Al ejecutar el programa aparecerá un menú que te preguntará si quieres «entrar código máquina», «examinar memoria» o «archivar bytes en cinta».

Si quieres entrar código máquina, pulsa 1. Después de fijar una dirección de comienzo y de pulsar RETURN, puedes empezar a introducir el código máquina. Este consiste en pares de dígitos en hexadecimal que debes ir introduciendo de dos en dos.

Puedes introducir todos los pares que quieras antes de pulsar RETURN,

pero es mejor teclear una línea cada vez y comprobar cuidadosamente los dígitos antes de pulsar RETURN. Es mucho más fácil editarlos mientras todavía están en la pantalla que cuando ya están en la memoria.

Observa que tus programas traducen a decimal los números que tú les das en hexadecimal, antes de hacer el POKE de los mismos a memoria. Esto se debe a que POKE es un comando del BASIC, y el BASIC de tu máquina no acepta numeración hexadecimal. No obstante, es mejor que pienses en hexadecimal, lo cual te dará una sensación más próxima a la forma de trabajar de la máquina.

Los programas en código máquina deben terminar con un signo ı, con lo que regresarás al menú inicial.

Si elegiste la opción 2 para examinar la memoria, el programa te pedirá de nuevo una dirección de comienzo; esta vez se trata del principio de la zona de memoria que quieres examinar. Si lo que quieres ver es todo el programa en lenguaje máquina, esta dirección de comienzo deberá coincidir con la dirección de comienzo de programa que le diste anteriormente.

Seguidamente el programa te preguntará si quieres una copia de impresora del programa en lenguaje máquina, en el caso de que tengas una. Si no quieres copia y pulsas N, el programa imprimirá la dirección de comienzo y el contenido de la misma, así como las siete direcciones siguientes de memoria.

Al pulsar cualquier tecla, excepto RETURN, aparecerán las primeras ocho direcciones junto con su contenido y el contenido de los siete bytes siguientes. Esto aparecerá en la línea siguiente, y de esta manera se puede imprimir, línea a línea, todo el programa en código máquina.

Si detectas algún error y quieres corregirlo, no tienes más que pulsar RETURN y el programa regresará al menú de partida. Pulsa ahora 1 para introducir código máquina y el orde-



TIMAS NOVEDADES... ULTIMAS NOVEDADES... ULTIMAS NOVEDADES... ULTIM

PREMIUM



SOFTWARE, S.

NAPOLES, 98, 1.º 3.ª - Tels. 232 24 61
08013 BARCELONA (SPAI

EL PODER DE L

PAUL McCARTNEY'S
Give my regards to
BROAD STREET
When the music stops, the mystery begins...
COMMODORE 64
MIND GAMES
Spectrum 48K

BROAD STREET

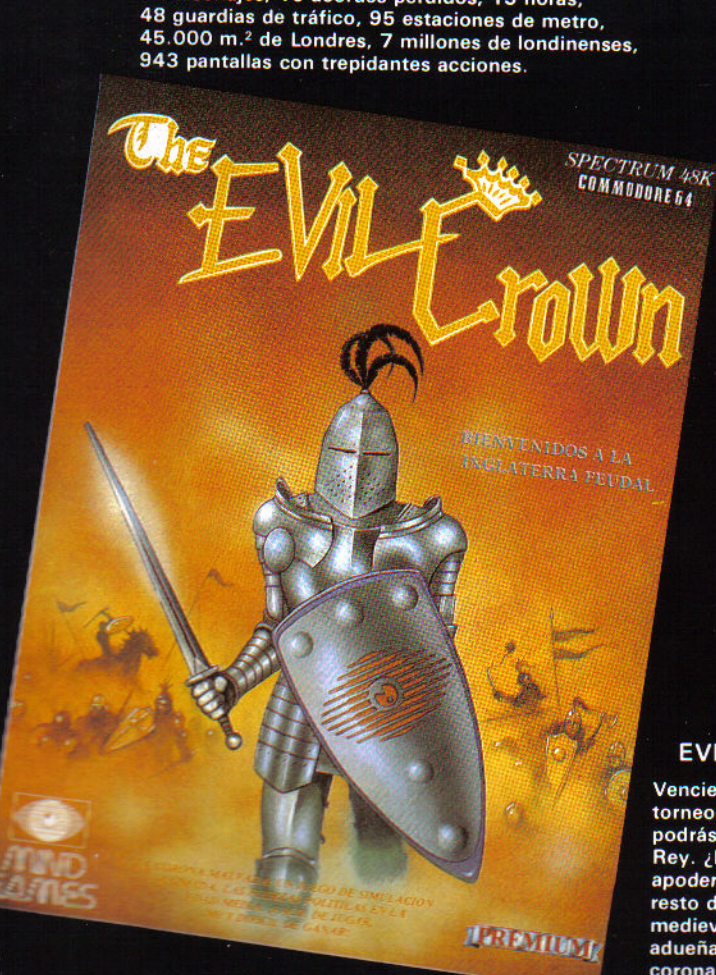
7 Personajes, 10 acordes perdidos, 15 horas, 48 guardias de tráfico, 95 estaciones de metro, 45.000 m.² de Londres, 7 millones de londinenses, 943 pantallas con trepidantes acciones.

SOLICITANOS TUS PROGRAMA

LASER ZONE (Spectrum)	975 Pts.
GRIDRUNNER (Spectrum)	975 Pts.
ASTRO BLASTER (Spectrum)	1.275 Pts.
FRENZY (Spectrum)	1.275 Pts.
FRIDGE FRENZY (Spectrum)	1.800 Pts.
QUINTIC WARRIOR (Commodore)	1.275 Pts.
STRONTIUM DOG (Spectrum)	1.800 Pts.
STRONTIUM DOG (Commodore)	2.100 Pts.

PURPLE T
BROAD S
BROAD S
EVIL CRO
EVIL CRO
TREASUR
TREASUR
ZAKIL WO

¡¡¡ATENCION!!! POWER, SOFTWARE, S
medios legales a su alcance, cualquier tipo de p
importaciones o duplicados ilegales que sobre su



EVIL CROWN

Venciendo en los torneos reales podrás llegar a ser Rey. ¿Podrás apoderarte del resto del mundo medieval y adueñarte de la corona malvada?

Strontium DOG
and the Death Gauntlet
TURBO LOAD
COMMODORE 64
+JOYSTICK
SPECTRUM 48K

STRONTIUM DOG

Johny Alpha es un agente investigador des
cazador de recompensas del futuro. Johny ll
electrobengalas, que deslumbrarán a sus e
obligándoles a dejar de disparar, y bombas
Buena suerte en tu misión.

MAS NOVEDADES... ULTIMAS NOVEDADES... ULTIMAS NOVEDADES... ULTIMAS

nador te pedirá una dirección de comienzo. Tienes que darle la dirección del byte erróneo. Si hay toda una serie de bytes que están mal, sólo tienes que dar la dirección del primero de ellos. A continuación puedes ir introduciendo los bytes correctos.

En el caso de que sólo se trate de un error aislado, puedes corregirlo «rompiendo» el programa monitor y un POKE con la corrección en la dirección adecuada. No te olvides sin embargo de que el POKE requiere un número decimal.

Cuando hayas terminado de introducir la corrección, pulsa RETURN, y a continuación examina de nuevo la memoria a fin de cerciorarte de que ha quedado hecha la corrección.

ARCHIVA TUS RUTINAS

Si quieres conservar el monitor de código máquina, tienes que archivarlo en cinta o disco por separado de la rutina en código máquina, en la forma usual. Con la opción de «archivar bytes en cinta» que te brinda el programa, sólo se archivan las rutinas en código máquina que han sido introducidas en este programa.

Al pulsar la opción 3, el programa te pedirá de nuevo una dirección de comienzo. Esta debe ser la dirección de arranque de la rutina en código máquina que acabas de introducir, aunque puedes archivar cualquier parte de la memoria utilizando diferentes direcciones de comienzo.

También se te pedirá una dirección final, que ha de ser la dirección de comienzo más el número de bytes que ocupa el programa. También puedes saber la dirección final examinando la memoria.

Después de que le hayas dado un nombre a la rutina, la máquina presentará una línea de programa en lo alto de la pantalla. Haz un HOME del cursor y pulsa RETURN tres veces. A continuación pulsa las teclas de PLAY y RECORD del magnetófono para archivar el programa de la forma acostumbrada.

Pero el monitor de código máquina no puede ser utilizado de nuevo hasta que los punteros que se modificaron durante la rutina de almacenamiento hayan sido definidos de nuevo. Para hacerlo, pon el cursor al principio del programa a mitad de pantalla y pulsa RETURN tres veces. Esto hará un POKE que restaurará los valores originales de los punteros y ejecutará el programa otra vez automáticamente.

CARGA DEL CODIGO MAQUINA

El comando LOAD para introducir código máquina tiene el siguiente formato:

```
LOAD "Nombrerutina",1,1
```

«Nombrerutina» es el nombre de la rutina en código máquina que quieres cargar. Siempre ha de ir entre comillas. El primer 1 es el número de dispositivo, 1 significa magnetófono. El segundo 1 ordena al **Commodore** que disponga el programa situándolo exactamente en la misma porción de memoria de donde vino.

Si efectúas la carga desde un disco, debes complementar la instrucción LOAD con 8,1.

Si alteraste alguno de los punteros del BASIC al escribir y archivar la rutina en código máquina, debes restablecer de nuevo los valores cada vez que se modifiquen por cualquier causa, por ejemplo el apagado de la máquina. En caso contrario tu rutina en código máquina no estaría protegida y podría ser que el BASIC se escribiera por encima de ella.

MIRANDO HACIA ATRAS

Las siguientes rutinas sirven para desplazar la pantalla hacia atrás, es decir de arriba a abajo en vez de abajo hacia arriba. Utiliza la rutina correspondiente a tu máquina con ayuda del monitor de código máquina.

Teclea para C-64 y Vic-20

```
A9 13 20 D2 FF A9 11 20
D2 FF A9 9D 20 D2 FF A9
94 20 D2 FF A9 A0 85 DA
A9 0D 20 D2 FF 60 #
```

EJECUCION DE LOS PROGRAMAS

Una vez tecleado el programa en código máquina y comprobada la ausencia de errores después de examinar la memoria, ha llegado el momento de ejecutarlo. Pero te encontrarás que no responde al familiar comando RUN, que es una isntrucción de BASIC y sólo sirve para ejecutar programas escritos en BASIC. Para ejecutar programas en lenguaje máquina, hacen falta instrucciones especiales.

La forma más fácil de ejecutar un programa en código máquina en el **Commodore 64** o en el **Vic 20**, es teclear SYS, seguido de la dirección de comienzo de la rutina en código máquina. Pero si quieres cambiar variables entre la rutina en código máquina y el BASIC, puedes utilizar la función USR del BASIC.

La sintaxis empleada es la siguiente:

Teclea para C-64 y Vic-20

```
A=USR(B) o PRINT USR(B)
en modo directo
```

en modo directo. USR envía al ordenador a la rutina en código máquina cuya dirección de comienzo está almacenada en las direcciones de memoria 785 y 786 (en el formato habitual byte bajo/byte alto). Por eso debes hacer en estas direcciones un POKE con las direcciones de comienzo de las rutinas en código máquina que deseas ejecutar en primer lugar para modificar los punteros.

El valor de B se pasa al programa en código máquina a través del primer acumulador de punto flotante, que ocupa las direcciones de memoria 97 a 102 (97 es el exponente, 98 a 101 es la mantisa y 102 contiene el signo). USR da también el contenido del acumulador de punto flotante cuando se termina la rutina en código máquina. De esta forma se puede utilizar para pasar valores desde la rutina en código máquina al programa en BASIC.

EL BOMBAZO del año : la primera tienda exclusivamente dedicada a COMMODORE :

Compuland

Corre a COMPULAND

Alli podras encontrar todo lo que las demas tiendas del pais no quisieron importar por considerarlo de un nivel superior a las necesidades de un usuario normal: nosotros no hemos cometido este error y por tanto tenemos lo que ningun otro puede ofrecer:

-Las ultimas novedades de SOFTWARE, traidas especialmente por nosotros y para nuestros clientes.

-Lo mas reciente en Hardware.

-Un unico y exclusivo CLUB DE SOFT .

-Y una atencion muy especial al disco y al 128.



NO LLORES

Hemos creado un CLUB DE SOFTWARE para que tengas acceso rapido y muy comodo de cualquier juego disponible en la tienda.

Asi mismo dispondras de la informacion mas relevante mediante una revista mensual ,que tratara de todos los temas de interes para el usuario que realmente este atraido por su micro.

Compuland dispone ya de la mayor seleccion de soft del pais, selecion que incluye tanto programas nacionales como importaciones directas de Gran Bretana y Estados Unidos.Cada mes se ira aumentando la biblioteca de programas disponibles importando todo lo que tenga un interes real para el usuario:

-Programas de diseno asistido por ordenador (tipo CAD).

-Programas musicales que te permitiran hasta la mezcla de voces.

-Programas inter-relacionados que combinan hoja de calculo, base de datos...



ASOMBRATE DEL HARDWARE DISPONIBLE:

-multiplicadores de velocidad de disco

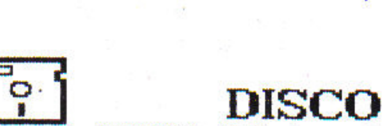
-Digitalizadores

-Samplers

-Tabletas graficas

-Taladradores de disco

-Lapices opticos de alta precision y un largo etc.....



DISCO

COMPULAND te ofrece ya una inmejorable gama de programas elegidos en las listas de numeros unos americanos e ingleses.Es ya una realidad y como tal se mantendra una perfecta correlacion con el soft de calidad (cualquiera que sea su pais de origen), tratando de ofrecer siempre una relacion calidad/precio lo mas ajustada posible:

-juegos
-utilidades
-herramientas

Y una sensacional oferta en DISCOS VIRGENES : LLAMANOS Y TE ASOMBRARAS.



NOS ENCONTRARAS EN LA CALLE CALVO ASENSIO N.8

TEL: (91) 243.16.38

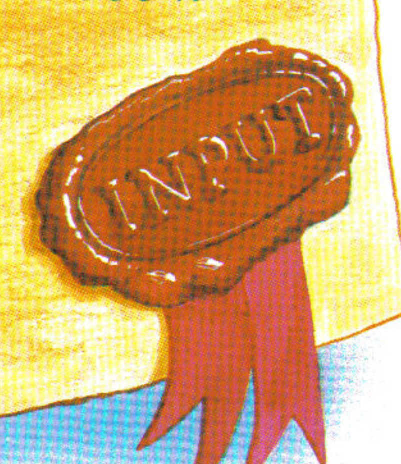


LOS MEJORES DE INPUT COMMODORE

PUESTO	TITULO	PORCENTAJE
1.º	Winter Games	20,4 %
2.º	Skyfox	17,6 %
3.º	Summer Games	10,5 %
4.º	Broad Street	11,2 %
5.º	The way of exploding fist	9,1 %
6.º	Misión Imposible	7,7 %
7.º	Karateka	7,1 %
8.º	Rambo	6,3 %
9.º	The Dambusters	5,6 %
10.º	Herbert's	4,5 %
		100 %

Para la confección de esta relación únicamente se han tenido en cuenta las votaciones enviadas por nuestros lectores de acuerdo con la sección «Los Mejores de Input».

Febrero de 1986



SOFTACTUALIDAD

Z, PARA MATAR MARCIANOS

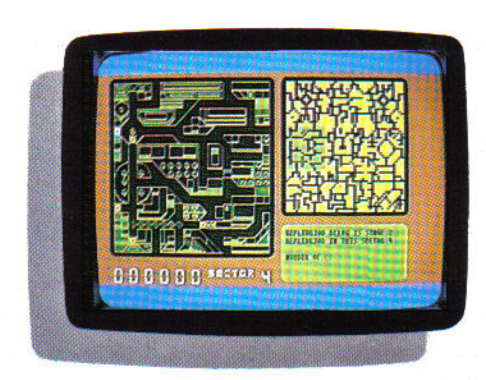
Z es otro juego de marcianos. Novedad inglesa del mes, que ha



llegado muy rápido al número uno de las listas de mayores ventas.
El argumento del programa no es demasiado original. Debes defender una ciudad del año 2000 contra los ataques de un ejército de naves tremendamente rápidas. Al igual que en *Transformers*, el *scrolling* es inmejorable y los gráficos superiores.
Tus reflejos son la clave.
La velocidad de las naves lo convierten en un juego difícil de controlar, y quizás sea este el único atractivo de un juego que no resalta precisamente por su originalidad.

TRANSFORMERS

Durante la pasada campaña de Navidad, hemos visto a menudo en la televisión los *robots* «Transformers» anunciarse una y otra vez. Esta novedad, también inglesa, es un juego de la casa **Ocean**, que lleva el mismo nombre del polifacético juguete. Deberás barrer y echar de la ciudadela a los mutantes que se han apoderado de ella. Dispones de 6 *transformers*, cada uno con características diferentes: fuerza, velocidad de vuelo, potencia de disparo...
Los gráficos evidencian un minucioso trabajo que desemboca en logrado



perfeccionismo. El *scrolling* es sensacional. Lamentablemente, los matamarcianos son un tema ya muy explotado.

MASTER KUNG FU

La casa **Data Est** resulta muy poco conocida para los usuarios, principalmente porque son pocos los programas que ha desarrollado. No sólo pocos, sino que a menudo se ha dedicado a la adaptación de



programas de un micro para otro.
Dos son los programas más recientemente adaptados para el **Commodore 64**: **Black Karate** y el **Master Kung Fu**.
Este último recrea las aventuras de un valiente karateka que ha de rescatar a su amiguita de un templo embrujado, en el que piso tras piso aumentan las dificultades.
Quizás os suene mucho más la versión del juego de las máquinas de los bares.
Se comenta que pronto este juego estará disponible en Europa gracias a **US-Gold**.

GANADORES DE LOS MEJORES DE INPUT COMMODORE

En el sorteo correspondiente al número 4 realizado entre quienes escribisteis mandando vuestros votos a LOS MEJORES DE INPUT han resultado ganadores:

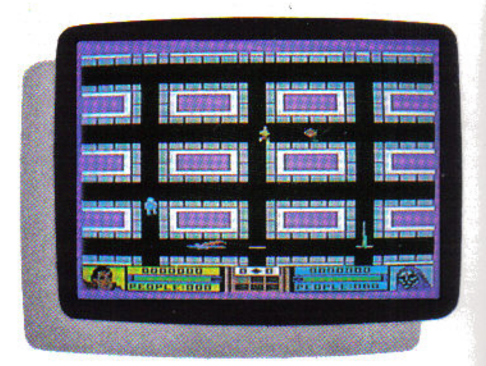
Angel Ancín Roldán (Pamplona)
Jordi Gómez Collado (Barcelona)
Pedro Altes Armengol (Tarragona)
Miguel Angel Piñero Sánchez (Pontevedra)
Agustín Gant Pérez De Siles (Córdoba)
Vicente Esquinas Fernández (Madrid)
Ricardo Sanz Lorenzo (Madrid)
Angel Alfonso Abril García (Valladolid)
Pedro García Fernández Valderrama (Madrid)
José Antonio Martínez Valenciano (Toledo)

SUPERMAN

Los habitantes de **Metrópolis** corren gran peligro. **Darkseid**, el astuto y malvado rival de **Superman**, intenta capturarlos; cree que en la mente de alguno de ellos está la fórmula Anti-Vida con la que podrá dominar el Universo. **Superman** intentará impedírselo, rescatando tantos «metropolitanos» como pueda antes que caigan bajo el poderío del ambicioso **Darkseid**. Para ello ambos rivales lucharán en el espacio, en las calles de la ciudad y en las prisiones subterráneas construidas por este diábolico enemigo del **Hombre de Acero**.

Darkseid utiliza el Rayo Omega para debilitar a **Superman** y para «teleportar» humanos. El **Hombre de Acero** no tiene poderes «teleportadores» pero puede volar y transportar con él a cualquier ciudadano. Posee, además, visión calorífica, superaliento y superfuerza para atacar a su enemigo o defenderse de él. Tanto la visión calorífica de **Superman** como el Rayo Omega de **Darksied**, proceden de sus ojos de forma que sólo se emiten en la dirección en que estén mirando y se activan apretando el disparador de tu *joystick*.

El juego se desarrolla a través de seis pantallas, que representan las calles de Metrópolis y las guaridas subterráneas. En ellas puedes encontrar deflectores, cuya disposición determinará la dirección de los rayos y los caminos a seguir por los habitantes. Hay además cinco



zonas de combate (cinco nuevas pantallas) donde los dos contrincantes son incitados a pelear: bolas de fuego, mortales rocas de *kriptonita*, rayos laser, y rayos omega son algunos de los obstáculos que **Superman** debe superar.

Tienes ahora la oportunidad de transformarte en **Superman** y ayudar a los habitantes de Metrópolis, pero si estás cansado de «ser bueno» y encarnar el papel del héroe, puedes asumir la personalidad del diábolico **Darksied**, tu ordenador ó tu contrincante se encargarán de hacer el resto.

Agota toda la fuerza de tu oponente ó rescata/captura el mayor número de ciudadanos, sólo así conseguirás vencer.

En el **Superman** de **First Star Software**, editado en España por Zafiro, el destino de Metrópolis y de sus habitantes depende de tí.

DATOS GENERALES	CALIFICACION
TITULO: Superman	GRAFICOS: 4 sobre 5
FABRICANTE: First Star	COLOR: 4 sobre 5
ORDENADOR: Commodore	PRESENTACION: 3,5 sobre 5
MEMORIA: 64k	INTERES: 3,5 sobre 5
CLASE DE PROGRAMA: Aventura fantástica	REALISMO: 3,5 sobre 5

EL ZOCO DE INPUT

Todo se compra y se vende. Los antiguos zocos fueron lugares destinados a todo tipo de transacciones. INPUT también tiene el suyo. Vuestras operaciones de compra, cambio o venta serán publicadas en esta sección, pero dos son las limitaciones que imponemos:

a) La propuesta tendrá que ver con la microinformática.

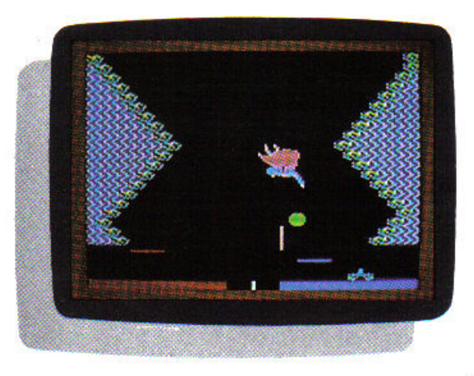
b) Nos reservamos el derecho de no publicar aquellos insertos de los que se sospeche un trasfondo lucrativo.

Ahora un ruego. Tratar de resumir al máximo el texto; escribir casi como un telegrama siendo claros y concisos.

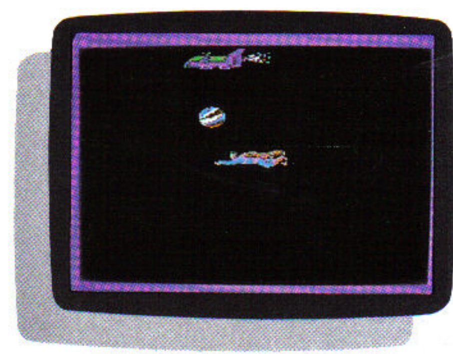


Envía tu mensaje a:

INPUT COMMODORE-ZOCO
c./ Alberto Alcocer, 46
28016 **MADRID**



Superman debe avanzar pese a los morfíferos disparos.



Todo tipo de sofisticadas armas son utilizadas contra el héroe.



Los espejos actúan como deflectores contra los rayos láser.

★★★★★★★★★★★★★★★★★★★★★★★★★

CONCURSO DE ONE-ON-ONE

Hace algunos números nos hacíamos eco de una convocatoria para que los comodoreros de Madrid pudieseis hacer gala de vuestras habilidades en baloncesto informático. Sin embargo, las firmas convocantes han tenido algunas dificultades técnicas para la puesta en práctica del aludido concurso.

One-On-One es un programa desarrollado por **Electronic Arts** y la firma **Dro Soft** es su representante en España y adaptadora de los programas al castellano. Es por ello que **Dro Soft** recoge el balón que estaba aún en el aire y nos dará a todos satisfacción y algunas sorpresas más.

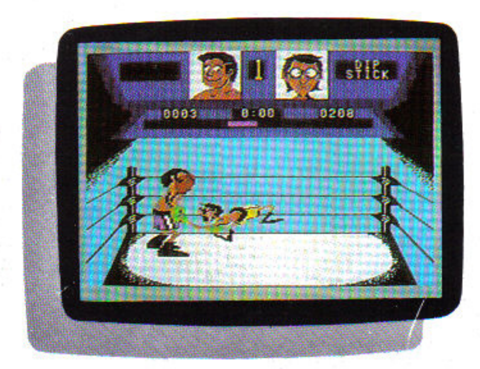
Por su parte, los premios en *hardware* los asumirá la prestigiosa firma **Micro Uno.**

La convocatoria definitiva para participantes y animadores será publicada en el número de marzo, para ese mismo mes. ¡Estar atentos!

NOCHE ESTELAR DE KID BYTE



A presenciar el evento acudirán hasta **Ronald Reagan** y **Margaret Tatcher**, por lo menos es lo que se deduce de la pantalla visualizada mientras se carga el programa, porque en ella aparecen ambos mandatarios.
En la entrada del estadio se agolpa el público para ver el combate nocturno del siglo. **Fight Night**, como su nombre indica es el último boxeo llegado de Gran Bretaña de manos de la harto conocida **US Gold**. La característica principal del programa es su inmensa flexibilidad.
Tienes total libertad para crear tu propio jugador: cabeza, brazos, piernas, torax, así como los colores que va a vestir. Con el monstruito que hayas creado podrás acudir al gimnasio, y a diferencia del **Larry Cugghan** de **Activision**, podrás entrenarte en vivo eligiendo velocidad, rotación y fuerza para los diferentes golpes, con tu palanca. De esta manera te será bien fácil recrear un boxeador con buenas características para el combate.
Tras la primera fase puedes pasar al menú general y elegir de entre dos



opciones: campeonato y liga. En esta última el nivel de tus contrincantes es más bajo, y por tanto vencer te costará menos.

Los oponentes están dotados de una particularidad: principalmente (a parte del aspecto que tienen), el golpe a traición que tratan de colocarte para derrumbarte. El primer oponente «**Dip Stick**» es francamente gracioso por el golpe bajo que en cuanto te descuidas te coloca: es un ser debilucho que no vacila en hacer trampas con tal de ganar. Los demás oponentes que irás encontrando están igualmente bien caracterizados (**Kid Castro**...), siendo de alabar el sentido del humor del programador.
El juego está cuidado al máximo en sus detalles (por ejemplo, el *ticket* de entrada al estadio aparece en pantalla mientras cargas al jugador). Los gráficos son excelentes y sólo se le puede reprochar pobreza en el sonido, claro que ¿acaso se oye música mientras se ve una pelea de boxeo?
Este es sin duda el mejor boxeo creado hasta la fecha, tanto por sus características como por la facilidad de manejo.

DATOS GENERALES	CALIFICACION
TITULO: Fight Night	GRAFICOS: 4,5 sobre 5
FABRICANTE: Sidney U.S. Gold	COLOR: 4,3 sobre 5
ORDENADOR: Commodore	PRESENTACION: 4,8 sobre 5
MEMORIA: 64K	INTERES: 4 sobre 5
CLASE DE PROGRAMA: Boxeo	REALISMO: 4,4 sobre 5

****

EL BYTE MAGICO

Steve Chapman, creador de **Quo Vadis**, ha lanzado un nuevo programa llamado **Wizardry**, en la línea de los juegos acción y aventuras que combina aspectos típicos de los juegos de estrategia. Según su autor, **Wizardry** es el mejor juego de «Dragones y Mazmorras» jamás creado. Quizás esta afirmación sea cierta, pues el juego tiene características muy poco comunes. Por una parte dispone de un fenomenal *scrolling* de pantalla, que permite recorrer todos los pasadizos y estancias hasta llegar a la Ciudad de las Ilusiones, elemento principal, y destruirla mediante un conjuro.

EL **BOMBAZO** del año : la primera tienda exclusivamente dedicada a COMMODORE :

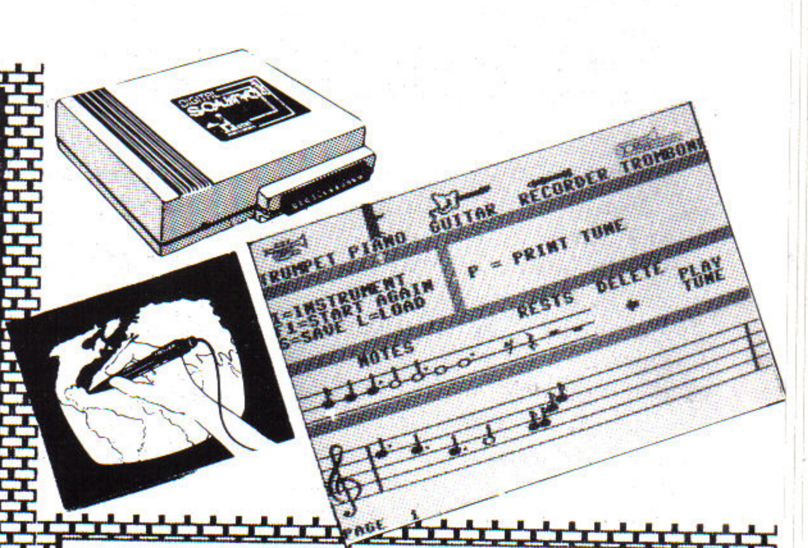
Compuland

SOFTWARE:

(JUEGOS DISPONIBLES EN CINTA Y DISCO)

- Ye har kung Fu
- Street hawk
- Enigma Force
- The Last V.8
- Knight Rider
- Friday the 13
- Starion
- Yabba dabba Do
- Cauldron
- Cauldron I I
- Rock & Wrestle
- Thunderbirds
- Blade Runner
- Critical Mass
- Transformers
- REVS
- Neverending st
- Nexus
- Little Comp. Pres
- Desert Fox
- Kennedy Appr.
- Zoids
- Koronis Rift
- Eidolon
- Roller Coaster
- Death wake ----

Y MUCHOS MAS



UTILIDADES:

- Wiza Star
- Mini Office
- Cobol
- Fortram
- Tasprint
- Taswork
- Soft 128
- Magpie data
- Wiza Write
- Azimuth
- Games Creator
- Machine tutor.
- Micro Illust
- Music System
- Mac Music
- Editores disco
- Doctor V.2
- Compiler 64
- Koala I I
- Pascal

Y MUCHO MAS...

HARDWARE:

- QUICK DISC multiplica el disco * 5, editor disco, DOS,Reset,Backup
- WISA STAR: por fin el LOTUS 1,2,3 para el commodore.
- SAMPLER: Digitalizador de sonido ultima novedad de las ferias GB
- PINZAS ESPECIALES para cortar discos y aprovechar a doble cara 320 kb / disco
- Cajas de ritmo, digitalizadores de video...
- Programas de dibujo con LAPIZ OPTICO a precios revolucionarios





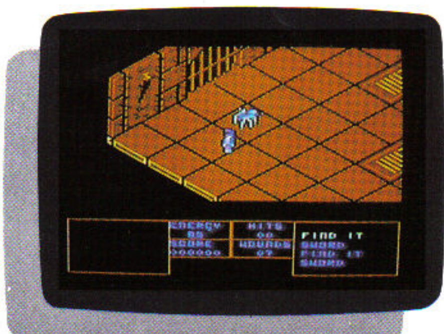


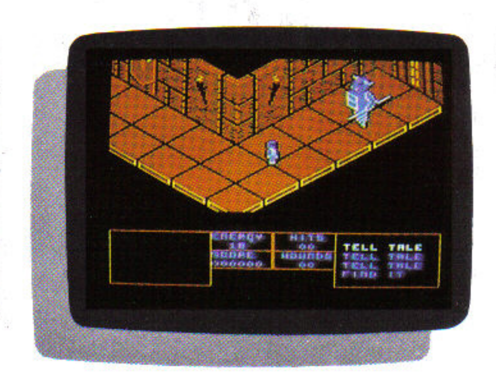


DATOS GENERALES	CALIFICACION
TITULO: Wizardry	GRAFICOS: 4,5 sobre 5
FABRICANTE: Edge	COLOR: 4,5 sobre 5
ORDENADOR: Commodore	PRESENTACION: 4,5 sobre 5
MEMORIA: 64K	INTERES: 4 sobre 5
CLASE DE PROGRAMA: Magia fantástica	REALISMO: 4 sobre 5

derecha de la pantalla. La tecla **Commodore** te permite abrir las puertas que aparecen en el recorrido. Para enfrentarte a los diversos peligros que se presentan (ogros, minotauros, globos y cajas que liberan gas mortífero) tendrás que utilizar los objetos y elementos que hallarás por el camino.
Quizás la característica más relevante de este magnífico programa sea la calidad gráfica: túneles tridimensionales y más de setenta áreas diferentes.







Las habitaciones son mudos testigos de los combates a espada (encontrarás muchas en el camino), encantamientos y peticiones de ayuda al maestro del lugar.
Desde las teclas 1 y 2 se escogen diversas opciones que se visualizan en una columna de la parte inferior

Otra relevante característica es el espléndido sonido ofrecido, con multitud de melodías diferentes que acompañan a cada oponente.

****

DISPARA FORASTERO

El tema de la **Ley del Oeste** es casi tan viejo como el Lejano Oeste mismo. En este juego eres un forastero que llega a un pueblo perdido en un verde valle de las montañas. Allí afrontas el desafío en duelo de la belicosa gente de la población. Has de diparar incluso contra una mujer de dudosa reputación, que te sale al paso, pero —eso sí— cuando muere cae al suelo con mucha gracia.

DATOS GENERALES	CALIFICACION
TITULO: La Ley del Oeste	GRAFICOS: 4,8 sobre 5
FABRICANTE: Acolade	COLOR: 4 sobre 5
ORDENADOR: Commodore	PRESENTACION: 4,5 sobre 5
MEMORIA: 64K	INTERES: 3,5 sobre 5
CLASE DE PROGRAMA: Desafío en el Oeste	REALISMO: 4,2 sobre 5

Tú, el jugador, apareces siempre de espaldas para desenfundar tu pistola, que puedes dirigir con el movimiento del *joystick* hacia el objetivo que desees alcanzar. Para ello dispones de un punto de mira que ajustará la puntería. Tanto el brazo como la pistola cambian de posición, para dirigir el cañón de la pistola hacia su blanco.
Se produce la clásica discusión entre el forastero y los distintos personajes que surgen del *saloon*, la estación, la cárcel, y el banco, etc.., a medida que las pantallas del paisaje de fondo van alternándose. En un momento dado desenfunda uno u otro y dispara el más rápido.
Los detalles de fondo, huidizos personajillos que corren a ocultarse y los temas musicales de acompañamiento proporcionan un ambiente adecuado para someternos a la **Ley del Oeste**.







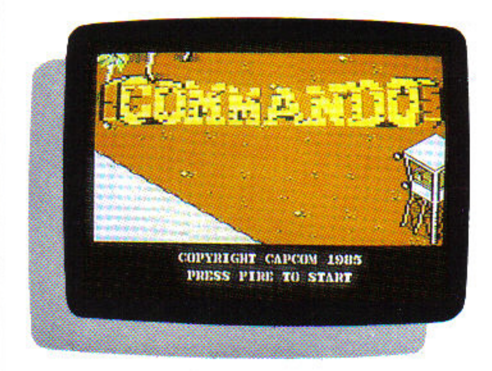
****

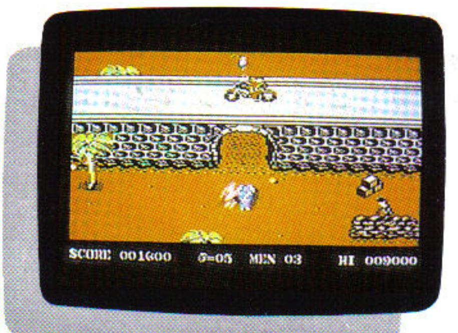
COMANDO SUICIDA

Una vez cargado este programa en el ordenador, te conviertes instantáneamente en un moderno soldado de élite de las fuerzas de choque.

Como es natural, te encomiendan una misión destinada a detener el avance de las fuerzas rebeldes, contando únicamente con una M60. Los rebeldes disponen de completos arsenales y no dudan en utilizar todo tipo de utensilios de guerra para aniquilarte. Debes responder con

DATOS GENERALES	CALIFICACION
TITULO: Comando	GRAFICOS: 4 sobre 5
FABRICANTE: Elite	COLOR: 4,3 sobre 5
ORDENADOR: Commodore	PRESENTACION: 4 sobre 5
MEMORIA: 64K	INTERES: 4,5 sobre 5
CLASE DE PROGRAMA: Aventura bélica	REALISMO: 4 sobre 5







toda la contundencia que permita el momento y sobre todo no retrocedas nunca.

Ah!, no olvides recoger las granadas que abandona el enemigo en su retroceso.

Cuando, alcances la fortaleza que utilizan como cuartel general, habrás conseguido el objetivo que te encomendaron. Recuerda que es importante no dejar nada peligroso en tu retaguardia, podría costarte caro.

****

BASEBALL

Hardball es el título de este completo juego de Baseball. Aunque este deporte no es demasiado conocido en nuestro país, seguro que pronto se hará popular entre los usuarios del **64**, porque aporta características de realismo tan sensacionales, que parece que nos encontremos ante secuencias de una película de dibujos animados: el efecto de movimiento es perfecto.

Como en la mayoría de los deportes competitivos, existen dos equipos uno vestido de azul y el otro de rojo.

El control de los jugadores se obtiene a partir de los dos *joysticks*.

El jugador que aparece de espaladas, el *pitcher*, lanza la pelota pudiendo elegirse el modo de disparo en

DATOS GENERALES	CALIFICACION
TITULO: Hardball	GRAFICOS: 5 sobre 5
FABRICANTE: Acolade	COLOR: 4,8 sobre 5
ORDENADOR: Commodore	PRESENTACION: 4,7 sobre 5
MEMORIA: 64K	INTERES: 4,5 sobre 5
CLASE DE PROGRAMA: Competición deportiva	REALISMO: 5 sobre 5







función del movimiento de la palanca, que puede ofrecer un tiro rápido, con trayectoria curva, etc.

El *catcher* puede ser igualmente controlado, para colocar el guante en la posición adecuada, en combinación con el lanzador.

Darle con el *bate* a la pelota en trayectoria es casi tan difícil como en la práctica.

Como en el juego real, las carreras de base a base con una constante, visualizándose una perspectiva de cada rincón del estadio. El jugador puede incluso derrapar tumbado en el suelo, para conseguir que el final de su carrera alcance el éxito.

El sonido de fondo, así como algunos pequeños detalles que observaréis, son el complemento que redondea el realismo al que aludimos.

****

PISA EL ACELERADOR... Y EL FRENO TAMBIEN

Revs es algo más que un programa de competición automovilista en Fórmula 3. Es todo un curso de aprendizaje para pilotos noveles. El **64** (o **128**) te sienta al volante de un fórmula japonés, el **Ralt RT3 Novamotor**, de **Toyota**. Su corazón es un motor de serie con dos mil centímetros de cubicaje. Dispone de los accesorios para que puedas ser el rey de los circuitos, si tus habilidades y reflejos como conductor lo permiten.

DATOS GENERALES	CALIFICACION
TITULO: Revs	GRAFICOS: 4,2 sobre 5
FABRICANTE: Fire Bird	COLOR: 4,5 sobre 5
ORDENADOR: C-64 C-128	PRESENTACION: 4,8 sobre 5
MEMORIA: 64K	INTERES: 4,5 sobre 5
CLASE DE PROGRAMA: Competición	REALISMO: 4,7 sobre 5

No cabe la menor duda de que toda esta clase de juegos recuerda al legendario **Pole Position**. Esta es una versión notablemente mejorada. El punto de vista se aproxima más al real, situandose a la altura de la vista de un piloto sentado en él. Una de las posibilidades que sorprenden más agradablemente es el abanico de opciones que le son ofrecidas al futuro conductor. Puede emplear las modestas teclas del ordenador, también el *joystick*, pero lo más







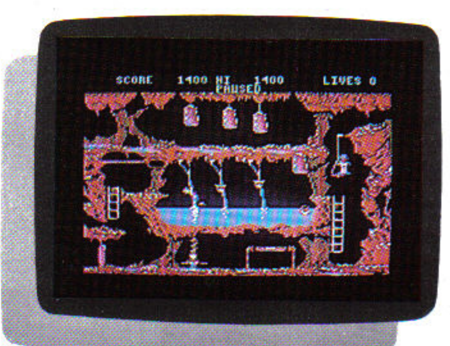
sorprendente es que podemos utilizar en combinación con ellos un *paddle* (*joystick analógico* = potenciómetro) con lo que obtenemos un control más realista del vehículo.
Se puede acelerar, frenar, cambiar las velocidades y hasta utilizar el embrague.
El manual que acompaña al juego está magnificamente editado y es un breve curso de técnicas de conducción deportiva.

★★★★★★★★★★★★★★★★★★★★★★★★★

LOS GOONIES

Datasoft, conocida casa de *soft* americana produjo este año tres programas de altisima acogida por parte del usuario: **Zorro**, **Goonies**, y **Alternative Reality**. El éxito alcanzado fué impresionante. El que se llevó la palma fué **Goonies**.
Goonies sigue en gran medida el guion de la película con **Mamma** buhardilla y lanzará dinero a la siniestra **Mamma** para entretenerla y conseguir así tiempo para bajar hasta la chimenea.
Pero quizás la característica más novedosa consista en que es la primera vez que un juego plantea una opción de dos jugadores participando al mismo tiempo, colaborando entre sí (en la opción de un jugador se cambia de un personaje a otro pulsando el disparador).
Para tener éxito con los trucos, deberás coordinar a la perfección a ambos individuos, para que los trozos de techo, los murciélagos o la misma **Mamma Fratelli** no te hagan añicos.
El nivel de dificultad de las pantallas es variable. Tal vez las más difíciles sean la pantalla del órgano y la del barco fantasma con su tesoro, donde





Fratelli, la gordinflona cascarrabias...,etc. Están presentes todos los personajes relevantes.
El juego consiste en pasar de unas pantallas a otras (en total son 8) y este es precisamente un gran atractivo del programa. Cada pantalla tiene un truco o varios, que has de encontrar para pasar a la siguiente. Por ejemplo, en la primera hay que lograr desaparecer por el hueco de la chimenea, pero uno de los participantes subirá a la

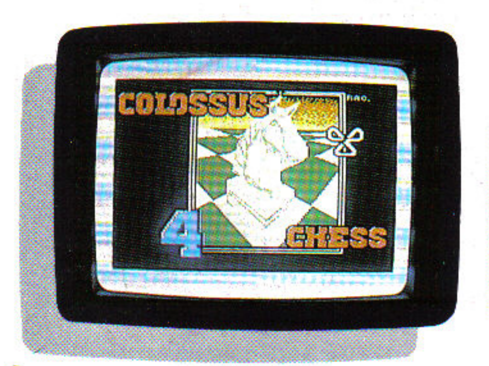
DATOS GENERALES	CALIFICACION
TITULO: The Gonnies	GRAFICOS: 4,5 sobre 5
FABRICANTE: Datasoft U.S.Gold	COLOR: 4 sobre 5
ORDENADOR: Commodore	PRESENTACION: 4,8 sobre 5
MEMORIA: 64K	INTERES: 5 sobre 5
CLASE DE PROGRAMA: Aventura de cine	REALISMO: 4,5 sobre 5

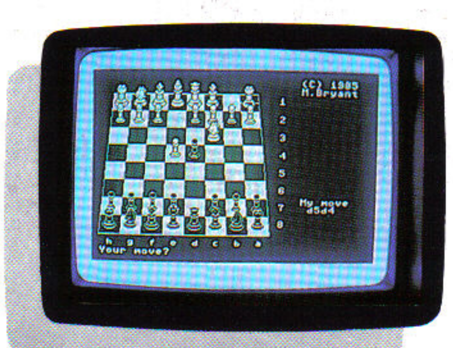
la **Mamma** está esperándote para obsequiarte con unos azotes. En cualquier caso te costará un poco progresar en el juego.
En varios aspectos, **Goonies** sigue la tendencia aparecida en los Estados Unidos de orientación del *soft* hacia terrenos más «inteligentes», en el que se mantiene la base de juego en torno al *joystick*, pero incorporando elementos de razonamiento.
Las excelentes características gráficas del juego y el sonido (que incluye la banda musical de la película) unidas a la opción de doble jugador y las facilidades de ir con la silla a cuestas, convierten a **Goonies** en un juego superadictivo.

****

COLOSSUS

Colossus Chess 4 es un nuevo juego de ajedrez presentado por **Zafiro** para el **Commodore 64**.
La presentación se verifica en dos pantallas. La principal muestra un tablero de ajedrez con la posición actual de las figuras. Las letras y números alrededor del tablero indican la anotación algebraica utilizada para cada cuadro. El tablero que aparece es el utilizado normalmente, pero **Colossus** ofrece la posibilidad de presentarlo en tres dimensiones. La segunda pantalla muestra el último registro de movimiento, los nombres de los jugadores y los relojes que miden el tiempo total que invierte cada contrincante en su partida.
Las fichas puedes moverlas con el *joystick* o mediante el teclado. El juego te brinda ambas maneras para desarrollar las jugadas. Debes elegir la más cómoda para ti.
También puedes elegir el color del tablero, de los bordes y de las anotaciones, entre los dieciséis disponibles. Pero no es esto todo lo que **Colossus** ofrece: modificar el tiempo establecido para cada jugador haciéndolo más corto o más largo; jugar al «ajedrez ciego», haciendo invisibles las piezas que desees, tal y como alguna vez han hecho los buenos jugadores de ajedrez; invertir la orientación del tablero, de forma que juegues con las negras arriba; dar marcha atrás en la partida hasta la jugada que desees y seleccionar la pieza en el ascenso o coronamiento de peón, son algunas de las extraordinarias posibilidades de este juego.
Y si eres un principiante o te cansas de jugar, puedes hacer que **Colossus** juegue contra sí mismo permitiéndote así, estudiar las distintas jugadas. Ah!, también puede ser el árbitro de la partida que disputes con otro oponente que no sea él.
Todas estas órdenes, y otras muchas que te ofrece el juego, son introducidas en el ordenador de forma muy sencilla. Basta presionar dos teclas para conseguirlo.
El programa tiene además seis opciones de juego que permiten diferentes modos y velocidades. Y, si lo deseas, puedes cambiar el volumen del sonido sin tener que ir al monitor de televisión, sólo ajustando el de los *beeps* del programa, del 0 (silencio) al 15 (máximo).
Los relojes están a cero, el registro de movimientos está limpio y se te dá la opción de mover primero. **Colossus** está preparado y esperándote. ¡Buena Suerte!





DATOS GENERALES	CALIFICACION
TITULO: Colossus Chess 4	GRAFICOS: 4,5 sobre 5
FABRICANTE: CDS Software	COLOR: 3,5 sobre 5
ORDENADOR: Commodore	PRESENTACION: 4 sobre 5
MEMORIA: 64k	INTERES: 4,5 sobre 5
CLASE DE PROGRAMA: Ajedrez	REALISMO: 4 sobre 5

NOTA PARA LOS LECTORES DE MURCIA

El número del pasado mes de diciembre no fue distribuido Murcia y su zona de influencia, debido a un lamentable accidente que sufrió el camión que transportaba los ejemplares.

Se ha realizado un nuevo envío y durante los próximos días podréis localizarlos en el kiosco. De todas formas, también podéis pedirlos por correo a nuestro servicio de ejemplares atrasados.



¡NO TE PIERDAS NI UN SOLO EJEMPLAR!

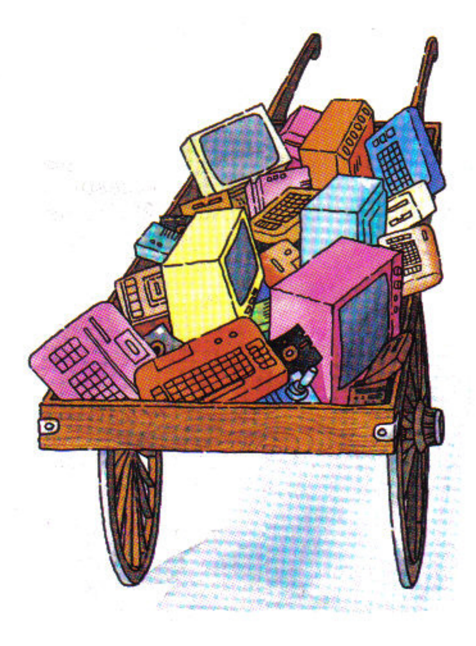
INPUT COMMODORE quiere proporcionar a sus lectores este nuevo servicio de ejemplares atrasados para que no pierdan la oportunidad de tener en sus hogares todos los ejemplares de esta revista, líder en el mercado español.

A partir de este mes podréis solicitar cualquier número de **INPUT COMMODORE** que querais, siempre al precio de cubierta (sin más gastos).

Utiliza el cupón adjunto, enviándolo a **EDISA** (Dpto. de Suscripciones), López de Hoyos, 141 - 28002 Madrid, o bien llámanos por teléfono al (91) 415 97 12.







EL ZOCO

Vendo Video-juego Atari.

Teléf. 786 03 79.
Antonio.

• • • • •

Estoy interesado en cambiar juegos para Commodore 64.

Preguntar por Juan.
Teléf. 242 98 21.
Barcelona.

• • • • •

Intercambio programas para C-64. Interesados escribid a:

Alfonso Carrasco.
Pza. San Jorge, 41.
Montcada-Reixac (Barcelona).

• • • • •

Compro Commodore 64 de segunda mano en buen estado. Ofertas a :

Salvador Larios Belmonte.
Industria, 6.
Teléf. 761 26 89.
Malgrat de Mar (Barcelona).

• • • • •

Vendo ordenador Amstrad CPC 464, monitor f.v., unidad de discos, e impresora Admate DP-100; osciloscopio de dos canales, generador B.F., polimetro, etc.; libros y revistas de electrónica. Pedir relación de material y precios.

Gonzalo Viejo González.
La Magdalena 24, 9º B.
Teléf. (985) 54 27 91.
Avilés (Asturias).

• • • • •

Vendo Commodore 64, datassette, programas de utilidad, manuales correspondientes, juegos. (1,5 año y en perfecto estado) 30.000 pts. (35.000 pts. con cartucho SIMONS BASIC ampliación excepcional del basic)

Teléf. (93) 237 56 36.

• • • • •

Vendo video-juego Philips videopac†G7400, 17.000 pts.

Bernardo Saez.
Egun-on,1 - 5º B.
Teléf. (943) 35 31 38.
Alza.
20017 San Sebastián.

• • • • •

Atención a todos los poseedores de un Vic-20, vendo memoria de 16K completamente nueva y con garantía por menos de mitad de precio. A quien lo desee, también tengo la primera y segunda parte de la Introducción al lenguaje de la programación, que por el precio casi las regalo.

Andrés Veá Baró.
San Andrés, 239 - 2º.
Teléf. (93) 345 49 02.
llamar de 7 a 10.
08030 Barcelona.

• • • • •

Vendo Commodore Plus/4 de Junio 85 † unid.1531 todo a 52.000 pts. con garantía, y apenas estrenado. Incluyo dos manuales de BASIC 3,5 y otro de 4 programas construidos en el mismo micro. 61,8K disponibles, o cambio por C-64 † C2N y una generosa cantidad de software o interfaces.

Carlos Felipe Spada.
Rosellón, 290.
Teléf. 257 75 51.
llamar al mediodía o por la noche.
08037 Barcelona.

• • • • •

Por cambio de equipo vendo el siguiente material: procesador de texto Easy Script (6.000.-), hoja electrónica Calc Result Avanced —cartucho y disco— (20.000.-), ampliación de basic Simon's Basic —cartucho— (8.000.-), sistema operativo bases de datos File-Boss 64 —cartucho y discos— (5.000.-), tableta gráfica Super Sketch —cartucho y pizarra— (10.000.-), programa 80 columnas con sprites y color (4.000.-), ordenador C-64 (35.000.-), monitor color con sonido Commodore 1701 (50.000.-). Todo el material es original y comprado a distribuidores oficiales de Microelectrónica y Control. Con el ordenador, el monitor o comprando varios de los otros elementos regalos diversos programas (sintetizador voz, simuladores vuelo, etc.). Perfecto estado.

Francisco Esquembre.
Molino de la Marquesa, 27.
Teléf. (96) 340 69 77 (casa).
horas trabajo 373 52 56.
Valencia.

• • • • •

Compro Commodore 64 de segunda mano en buen estado y precio.

Antonio Martín Medina.
Barcelona, 86.
Teléf. (972) 21 20 95.
17001 Gerona.

• • • • •

Vendo para el Vic-20 los cartuchos: Ayuda al programador, Road Race, Radar ratrace, Sargon II Chess. Además, la guía de referencia. Regalo programas.

Josep Capdevila.
Major de Sarriá, 52 - 3º 1.
08017 Barcelona.

• • • • •

Atención: me interesaría programa contabilidad para C-64, ofrezco juegos a cambio. También intercambio juegos.

Albert Casellas.
Carme, 9.
Igualada (Barcelona).

• • • • •

Compro C-64 con cassette o bien Amstrad 464 y/o monitor de fósforo (preferentemente de colores). Precio a convenir. Haced vuestras ofertas.

Jesús García Martínez.
Travesía de la Fuente, 1.
Navatejera (León).

• • • • •

Vendo Oric-1 (64K) con transformador, cables necesarios y dos juegos originales todo por 30.000 pts. Muy poco usado.

Jordi Puértolas Ramón.
Bl. San Marcos C - 2º, 2ª.
Teléf. 22 60 80.
horas de comer.
S. Pedro y S. Pablo (Tarragona).

• • • • •

EL ZOCO

Si te interesa conseguir un Commodore Vic-20 con un mes de uso, el datassette junto a un manual que ampliamente te describe el manejo y algunos programas; todo por 25.000 pts. comunícate conmigo.

Juan Manuel Tirado Pérez.
Avda. de la Paloma, 40.
Teléf. 31 84 55.
29003 Málaga.

• • • • •

Vendo ordenador LASER 310, teclado profesional, completísimo libro de instrucciones más un gran número de cintas con software. Todo aún para estrenar por tan sólo 25.000 ptas.

Teléf. (93) 352 76 41
(Horas de oficina).

• • • • •

Cambio cartucho Vic-Avenger por otro cartucho cualquiera que no sea el del ajedrez.

Toni Farras Pique.
Carretera de Lérida, s/n.
Orgaña Pro (Lérida).

• • • • •

Se vende Commodore Vic-20 precio a convenir según posibilidades económicas. Alrededor de las 22.000 pts. pero incluyendo curso de Basic, primera parte , juego original Ratman, cartucho de juego de marcianos, cartucho ampliación a 16K y un interface para conectarlo con cualquier cassette (precio real 50.000) ideal para iniciarse en la informatica.

Juan M. Robles Arans.
Dc Trueta, 31-33 - 2ª, 3ª.
Castelldefels (Barcelona).

• • • • •

Si dispones de todos o de algunos de los 7 primeros números de la revista Commodore World, y no los quieres, ponte en contacto conmigo.

José Antonio Flores.
Prim, 22.
Peñarroya-Pueblonuevo (Córdoba).

• • • • •

Me gustaría intercambiar programas para el C-64. Mandar lista.

Angel Martínez Domingo.
Canço, 1, 3.º - 2.ª.
08035 Barcelona.

• • • • •

Vendo C-16, en perfecto estado, con garantía 6 meses. Todos los accesorios disponibles más el cassette. 25.000 pesetas.

Gil Andrés.
Teléf. (985) 38 07 83.
(De 13.30 a 14.30
o a partir de las 20 h.).

• • • • •

C-64, intercambio de programas en disco y en cassette en Turbo.

Jorge Mediavilla Diez.
San José, 3 - 1.º dcha.
Teléf. (988) 72 29 32.
34004 Palencia.

• • • • •

Intercambio programas de todo tipo para el CBM-64, preferentemente en disco. Estoy sumamente interesado en programas de quinielas que hagan reducidas al 13. Poseo más de 800 programas para intercambiar; admito también intercambio para el 128.

F. M. López González.
Puerta Canseco, 47 - 3.º B.
38003 Sta. Cruz de Tenerife.

Ayúdanos a ayudarte

Puedes ganar un COMMODORE 64

Encuesta para conocer tu perfil y preferencias

Es muy difícil que podamos hablar con todos vosotros para conoceros mejor y saber cuáles son vuestras preferencias. Para tener constancia de opiniones y necesidades hemos de recurrir a las frías encuestas y dejar que se pronuncie la estadística. Los beneficiados seremos todos: vosotros porque nos decís qué queréis y nosotros porque al haceros caso continuaremos haciendo vuestra revista favorita.

1. ¿Tienes ordenador? Sí ☐ No ☐
2. En caso afirmativo. ¿Cuál? ....
3. ¿Cuánto tiempo hace que lo posees? ....
4. ¿Qué periféricos y accesorios tienes? ....
   ....
5. ¿Desde qué número lees INPUT? N.º: ....
6. ¿Cómo conociste INPUT? ....
7. ¿La compras en kiosco? ☐ ¿Eres suscriptor? ☐
8. ¿Con qué frecuencia la adquieres?
   Todas ☐
   De vez en cuando ☐
   Otros ☐
9. ¿Cuántas personas leen tu ejemplar de INPUT? ....
10. ¿Lees toda la revista? Sí ☐ No ☐
11. ¿Tecleas?:
    Todo ☐
    Casi todo ☐
    Alguna cosa ☐
    Nada ☐
12. ¿Califica qué secciones te gustan más con mayor puntuación (10 a 0):
    Programación ....
    Código máquina ....
    Robótica ....
    Coleccionable ....
    Revista de *hardware* ....
    Revista de *software* ....
    Aplicaciones ....
    Actualidad ....
    Libros ....
    Buzón ....
    Zoco ....
    Educación ....
    Los mejores de INPUT ....
13. Califica las portadas entre 10 y 0, según lo que te hayan gustado:
    N.º 1 ....
    N.º 2 ....
    N.º 3 ....
    N.º 4 ....
    N.º 5 ....
    N.º 6 ....
14. ¿Qué artículo te gustó más del ejemplar?:
    N.º 1 ....
    N.º 2 ....
    N.º 3 ....
    N.º 4 ....
    N.º 5 ....
    N.º 6 ....
15. ¿A qué te dedicas cuando no estás con el ordenador?, ¿estudias o trabajas?, ¿en qué, si no es indiscreción?):
    ....
16. ¿Qué secciones o temas echas de menos en INPUT?:
    ....
17. ¿Estás satisfecho con tu ordenador? Sí ☐ No ☐
18. ¿Estás contento con los servicios técnico y postventa? Sí ☐ No ☐
19. ¿De qué marca y modelo será el próximo ordenador que compres? ....
20. ¿Piensas cambiarlo pronto? Sí ☐ No ☐
21. ¿Aproximadamente, cuántos programas comerciales forman tu biblioteca? ....
22. ¿Cada cuánto tiempo compras uno nuevo? ....
23. ¿Lees otras revistas de ordenadores? Sí ☐ No ☐
24. En caso afirmativo,
    ¿Cuáles? ....
    ....
25. ¿Por alguna razón en particular? ....
    ....
26. ¿Quiéres añadir alguna cosa más? ....
    ....
    ....

Con solo contestar a esta encuesta puedes ganar un COMMODORE o su valor equivalente en accesorios y/o *software*..
Así, que escribe tus datos para participar. La fecha límite es el próximo 5 de abril. No es preciso que contestes las preguntas para las que no tengas respuesta. Tampoco es necesario que las respuestas sean favorables a nuestra labor para que puedas participar en el sorteo. Critícanos si honestamente crees que debes hacerlo. Envíanos este cuestionario o su fotocopia a: ENCUESTA INPUT COMMODORE, c/ Alberto Alcocer, 46-4.º 28016 MADRID.

Nombre .... Apellidos ....
Edad .... Dirección .... Dto. Postal ....
Teléfono de contacto .... Gracias y suerte con el premio.



LA LEY DEL OESTE

Si siempre quisistes llevar una correa con seis balas un alfiler con una estrella prendida en tu pecho, cerillas para realizar hazañas como Bat Masterson o Wyatt Earp, entonces la Ley del Oeste es tu eleccion. Olvida los disparos calle abajo en la calle principal. Es el año 1870 y tu eres el sheriff de Gold Guch. Tu no podrás sobrevivir a una ráfaga en el camino si en cada situación comprometida alguno de los pistoleros por un momento es demasiado rapido. Por eso usa las calles astutamente y trata de averiguar el "mal revolver" "mal disparo" de estos personajes. También debes de conservar los locales alegres. El salón el chico e incluso a tu propio sustituto. Todas son reglas que debes respetar si quieres estar vivo a media noche.

Fabricado y distribuido bajo licencia por:

COMPULOGICAL S.A.

Santa Cruz de Marcenado, 31 - 28015 Madrid - Teléf. 241 10 63

Distriduido en Cataluña y Baleares por:
DISCLUB, S.A. - Balmes, 58 - BARCELONA
Tel: (93) 302 39 08 - **P.V.P. 2.300 Ptas.**